

TU MICRO COMMODORE
nº 16 160 ptas (IVA incluido)
SEMANAL
BREAKDANCE:
LA FIEBRE DEL BAILE
ACOPLAMIENTO SIDERAL TECLA A TECLA
ESCRITOR: LA NUEVA MAQUINA DE ESCRIBIR
CHISPAS: TRUCOS PARA TODOS Y PARA TODO



Director:
ANTONIO M. FERRER ABELLO

Redacción:
ANTONIO CARVAJAL
JUAN M. LOPEZ MARTINEZ
PABLO GARCIA MOLINA
IGNACIO BARCO LUENGO
ALFEDO SINDIN VALERO

Colaboradores:
JOSE LUIS VAZQUEZ DE PARGA
ANTONIO MANZANERA

Secretaria de Redacción:
PILAR MANZANERA AMARO

Diseño y Maquetación:
CARLOS GONZALEZ AMEZUA

Ilustraciones:
ANTONIO PERERA
RAMON POLO

Fotografía:
Equipo Gálata

Directora Publicidad:
CARMINA FERRER
Tel. (91) 457 69 23

Publicidad Barcelona:
ISIDRO IGLESIAS
Avda. Corts Catalanes, 1010
Tel. (93) 307 11 13

Director de Producción:
VICENTE ROBLES

Directora de Administración:
MARIA ANTONIA BUITRAGO

Suscripciones:
MARIA GONZALEZ AMEZUA

Redacción, administración, publicidad y suscripciones:
Plza. República Ecuador, 2.
28016 MADRID. Tel. 250 58 20
Télex 49371 ELOC E

Dirección para correspondencia:
Aptdo. de Correos 61.294
28080 MADRID

Fotocomposición:
VIERNA, S. A. MADRID

Fotomecánica:
RODACOLOR, S. A. MADRID

Imprenta:
GRAFICAS REUNIDAS, S. A. MADRID

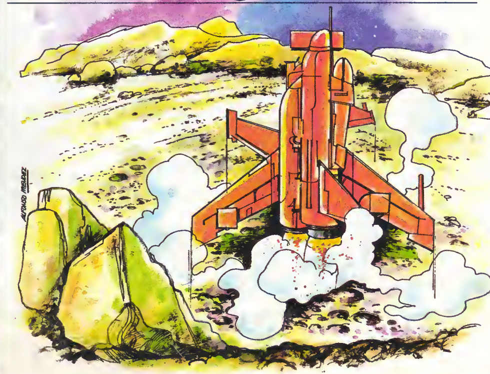
Distribución:
COEDIS, Valencia, 245.
BARCELONA

Precios para España. Ejemplar: (160 ptas. (IVA incluido). 155 ptas. (Canarias, Ceuta y Malilla). (La suscripción anual incluye 52 números.)
Distribución Cono Sur:
CADE, S.R.L.
Pasaja Sud América, 1532
Tel. 21 24 64.
Buenos Aires 1.290. Argentina.

Depósito Legal: M. 40920-1985.
Impraso en España.

SUMARIO

Como ya habíamos anunciado anteriormente, no es nuestra intención realizar únicamente el estudio A FONDO de programas de juego, sino también de aquellos destinados a tareas más serias. Buen ejemplo de ello es este número, en el cual sometemos a nuestro banco de pruebas al programa ESCRITOR de Casa de Software; una aplicación muy importante, no sólo por su calidad y por la misión que cubre: el proceso de textos (¡y que nos pregunten a los redactores si es importante un procesador de textos!), sino también porque es un programa adaptado a todas nuestras necesidades idiomáticas (castellano, catalán y francés), con acentos, eñes, etc... Merece la pena.

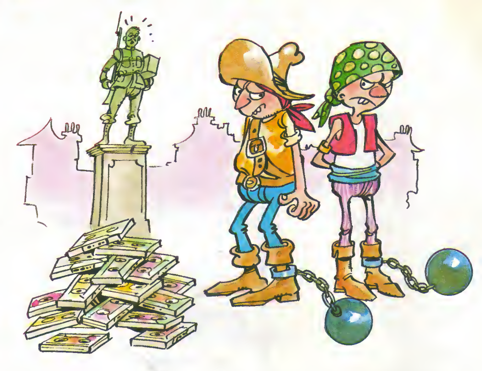


NOTICIAS

¡BUENAS NOTICIAS! SIGUE LA LUCHA CONTRA LA PIRATERIA

Ampliando nuestra información aparecida en el último número, procedemos a extractar el comunicado enviado a nuestra Editorial por la Oficina de Prensa de ANEXO, asociación a la cual nos hallamos adscritos, y que tiene por objetivo la supresión de cualquier actividad de piratería en nuestro país.

Tras la denuncia presentada por ANEXO (asociación española de empresas fabricantes de software), ante la creciente actividad de los piratas de programas, el Grupo Séptimo de la Brigada Regional de la Policía Judicial procedió el domingo, día 2 de marzo, a la retirada de cintas con programas para microordenador, que eran vendidas de forma ilegal en el «rastro» madrileño.



El número de programas incautados supera las 11.000 unidades, con un valor en el mercado de más de ¡22 millones de pesetas!, tomándose declaración a 28 personas presuntamente implicadas en la apropiación ilícita de los derechos de autor de dichos programas.

El tipo de programas pirateados es el medio de subsistencia de pequeñas empresas, que invierten considerables sumas de dinero, basándose en la venta de gran cantidad de dichos productos a un bajo precio. La piratería, practicada por grupos altamente profesionalizados, hace peligrar la naciente industria del software en nuestro país, y es por tanto un problema importante y nada desdeñable.

Hemos de tener en cuenta a la hora de juzgar este fenómeno, que un programa profesional para ordenador conlleva una inversión de unos 20 millones de pesetas, debido a lo cual, las actividades de sabotaje contra su venta legal, pueden dar al traste con cualquier economía, por fuerte que esta sea, generando los consiguientes problemas económicos y de empleo para el sector.

La prueba de la importancia de este «negocio», subterráneo e ilícito, se obtiene de la creciente profesionalización observada en los grupos dedicados a la comisión de este delito.

El equipo de nuestra redacción ha podido constatar personalmente, en la fuente de la noticia, cómo el pasado domingo, día 9 de marzo, varios individuos se encontraban apostados en las esquinas de las zonas de venta clandestina en el «rastro» madrileño, y auxiliados por Walkie-Talkies llevaban a cabo su misión de vigilancia, protegiendo las acciones de sus compañeros, que en la más completa impunidad continuaban la venta ilegal de programas, prestos al menor aviso, para camuflar sus puestos con unas persianillas, convirtiéndolos en unos inofensivos tenderetes de reventa de comics.



SUSCRIPCION DE LA MANO DE LOS 7 MAGNIFICOS

El agraciado ganador de una suscripción por un año a nuestra revista, por su colaboración en la confección de Los 7 Magníficos, ha sido nuestro amigo de San Sebastián:

José M.ª Arruabarrena Beraza Bustinchulo, 29, 3.° izq.

Enhorabuena a ti José María, y ya sabeis, los demás animaros a enviarnos vuestros votos, os esperan muchas otras oportunidades de haceros con el premio.



PROCESADOR DE TEXTO

Miguel de Cervantes Saavedra escribió una de las obras cumbre de la literatura universal: Don Quijote de la Mancha. Para ello, no utilizó un procesador de textos, pero es innegable que le hubiera resultado de gran utilidad. Dentro de la gama de software para Commodore 64 y 128, encontramos algunos procesadores de textos, si bien el programa ESCRITOR es el único preparado para los idiomas castellano, catalán y francés.



A, E, I, O, U... ¡Cuántas veces habremos repetido esta sucesión de letras en nuestra más tierna infancia!, ¿y todo esto para qué? Para aprender a leer y a escribir, nos decía incesantemente aquella profesora de ojos azules y rubios cabellos. Y nosotros, erre que erre, a con a, e con e..., aprendimos la lección y empezamos a desenvolvernos en el fantástico mundo del lenguaje, de la caligrafía, de la literatura...

Más tarde, al ver un ordenador personal utilizado como una máquina de escribir, surgió en nuestra mente la idea de emplearlos para uso propio, aprovechando todas las ventajas que ofrecen: muchas y muy variadas. El programa ESCRITOR, un procesador de textos presentado en cartucho, requiere únicamente un Commodore 64 ó 128, un monitor o receptor de televisión y una impresora; aunque también resulta casi imprescindible el uso de una o dos unidades de disco y/o una unidad de casete, con el fin de conservar el resultado de nuestro trabajo y realizar sucesivas correcciones sobre éste.

Multitud de opciones dependen de la acción de nuestros dedos sobre el teclado: desde reescribir el Quijote, hasta crear cartas personalizadas. Gracias a este paquete de software, los escritores (aficionados o profesionales) contarán con una ayuda insustituible, los estudiantes podrán entregar sus trabajos muy bien presentados, los oficinistas se ahorrarán muchos esfuerzos, las casas de software diseñarán los manuales de instrucciones de sus productos, y nosotros, los redactores de TU MICRO COMMODORE, escribi-

remos nuestros artículos (como en este caso).

FUNCIONANDO CON EL PROGRAMA ESCRITOR. EL PRIMER PASO: COMO MANEJAR EL TECLADO

Rompemos la funda de plástico que protege el procesador y el manual de instrucciones, observando detenidamente el cartucho. Ante todo, es preciso conocer el lugar donde vamos a insertarlo: EL PORT DE EXPANSION (orificio frontal derecho del ordenador). Para comenzar a escribir DEBEMOS APAGAR EL ORDENADOR, introducir el cartucho con la etiqueta mirando hacia arriba y después encender nuestro Commodore. Una breve pantalla de presentación y ¡tachan, tachan!, el programa ESCRITOR empieza su actuación y nosotros, la nuestra.

La pantalla consta de 24 columnas y 25 filas, de las cuales la primera será informativa y la escritura se verificará en las restantes 24. En la línea informativa encontramos, además del nombre del programa (ESCRITOR 1.01), seis datos de izquierda a derecha: la letra E o C muestra si el editor está en modo de escritura o comando, la letra M o un espacio revela el estado de escritura (mayúsculas o minúsculas), un 1 señala la utilización del casete como periférico y un 8 el uso de la unidad de disco, la C señala visualización continua del escrito, y los restantes valores corresponden respectivamente al número de línea y de columna donde se encuentra el cursor.

El número máximo de columnas (y consecuentemente de filas) depende del valor asignado a la X al pulsar las teclas **F7** y **A** (antes de escribir, pues este comando borra el texto en memoria), el cual varía desde 1 hasta 4. Un 1 selecciona 40 columnas, para un 2 hay 80, a un 3 le corresponden 96 y el 4 ofrece 137. Inicialmente, la X es igual a 2, pero con la finalidad de escribir cómodamente y tener siempre a la vista 24 líneas consecutivas del texto, asignaremos un 1 a esta variable.

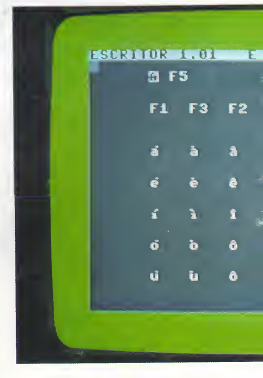
El teclado funciona de un modo normal exceptuando unas cuantas funciones supletorias añadidas por los diseñadores de la utilidad, tales como acceder al modo disco, acentos, eñes y muchas otras. En cuanto a la pantalla de escritura, cabe destacar su similitud con una ventana que se desplazará por encima del texto, permitiéndonos ver un fragmento. Entre la pantalla de escritura y el resultado del proceso de impresión de un documento existen muchas diferencias, motivadas fundamentalmente por las instrucciones de formato, definibles a nuestro antojo.

Casi todo teclado funciona normalmente; sin embargo, las teclas de función desarrollan cometidos específicos como la acentuación (**F1, F2, F3** y **F4**), comandos de formato (**F5**), modo disco (**F6**), modo comando (**F7**) y cambio de mayúsculas/minúsculas (**F8**).

Para acentuar las cinco vocales del vocabulario castellano, el procedimiento a seguir coincide con el de las máquinas de escribir, es decir, exige pulsar primero la tecla correspondiente al acento y a continuación la vocal en cuestión. Con **F1** aparecerá el acento agudo "´", con **F3** el grave "`", **F2** el circunflejo "^" y la diéresis "¨". No obstante, el procesador de textos contiene otros caracteres propios de la lengua castellana, catalana y francesa: la 'ñ' (**CBM + £**), 'Ñ' (**SHIFT + £**), 'ç' (**CBM +** ⊡), 'C' (**SHIFT** + ⊡), '¡' (**CBM + 1**), '¿' (**CBM + 2**) y 'Pt' (**CBM + 6**).

Asimismo, dos teclas más se sustraen del comportamiento normal del editor de pantalla del Commodore: **CLR** y **HOME.** La primera sitúa el cursor en la fila 1, columna 1, mientras que la segunda sube el cursor hasta el principio de la ventana de pantalla en la cual nos encontramos. Otra anomalía se presenta al pulsar las teclas **SHIFT + RETURN,** en cuyo caso habremos escrito un punto y aparte y aparecer una flecha hacia la izquierda en vídeo inverso. Por último, al pulsar la tecla superior izquierda, el cursor se desplazará hasta el último caracter de la línea anterior a aquella en la cual se encontraba situado.

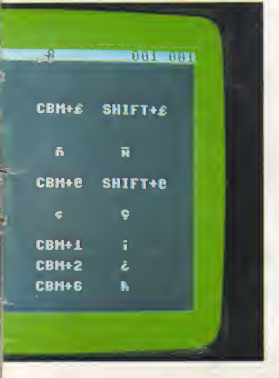




EL MODO COMANDO, REQUISITO INDISPENSABLE DE TODO PROCESADOR CON CATEGORIA

Conocemos ya una forma muy sencilla de introducir texto, aunque contamos con la ayuda de una serie de potentes comandos (accesibles PREVIA PULSACION DE LA TECLA F7) destinados a facilitar esta tarea enormemente: el modo comando. Al entrar en este modo, la línea informativa cambiará la E por una C, con lo cual evitaremos muchas confusiones.





Después de introducir en memoria un documento de prueba, efectuamos una copia de seguridad del mismo en un periférico. Al entrar en funcionamiento, el ESCRITOR trabaja con unidad de discos; el cambio a unidad de casete se realiza pulsando **F8.** Para grabar el fichero basta la tecla **G,** con lo cual el ordenador nos insta a introducir el nombre del fichero. Análogamente, la tecla **C** faculta la extracción de un fichero grabado, a partir de la posición del cursor.

El primer comando a utilizar es aquel que ofrece un ancho de texto de 40 columnas (para esto hemos hecho la copia de seguridad, porque el comando **A** borra toda la memoria textual). Si no apreciamos la forma de eliminar letras del ordenador, siempre podemos hacer uso de la tecla **E,** y las subopciones inherentes a ella: **T** elimina todo el texto, **R** provoca la desaparición de todas las líneas que se encuentren por debajo del cursor, y **A** borra un área predefinida.

Un área es un espacio o conjunto de líneas determinado por el primer caracter y el último. Para definirla situamos el cursor sobre el principio del área y presionamos **D;** colocándolo a continuación sobre el final y pulsando de nuevo **D.** Una vez realizado el proceso, con la **M** cambiamos la localización del área (se borra el área anterior); y con **R** realizamos una copia (el área anterior permanece en su sitio).

La búsqueda y/o sustitución de una palabra a lo largo de un texto se lleva a cabo mediante tres nuevos comandos. Tras posicionar el cursor en el origen (001, 001), la **B** inicia el proceso pidiendo el conjunto de caracteres a localizar. Si sólo nos interesa conocer el lugar que este ocupa, presionamos **RETURN** y proseguimos con el comando **L.** En caso de desear un cambio, señalamos una nueva cadena de caracteres y aplicamos el comando ***.** Tanto para la búsqueda como para la sustitución indicamos el sitio donde tendrá lugar el proceso (M = memoria, E = ficheros enlazados), y lo controlamos mediante la **C.**

Los tabuladores destinados a encolumnar caracteres de una forma muy peculiar. Los tabuladores normales (**T + N**) sitúan el último caracter de una cadena alfanumérica justo en el punto tabulado, no así los decimales cuya misión consiste en depositar los puntos decimales de las cadenas numéricas en el punto tabulado. Los tabuladores se graban con los ficheros, se visualizan con el comando **/,** y son borrados al pulsar **Q** sobre la columna de tabulación.

Las demás funciones son: **S** selecciona el número de línea donde nos interesa posicionar el cursor, **INST** abre un espacio dentro de una fila, **DEL** borra un caracter y desplaza el texto hacia la izquierda, e **I** establece el modo de inserción, muy útil para intercalar áreas, textos grabados, etc.

LOS COMANDOS DE SALIDA CONSTITUYEN EL PASAPORTE DE LOS TEXTOS EN MEMORIA

El único comando aún no estudiado quizá sea el más importante de todos, pues se encarga de mostrar el texto en pantalla o en impresora: el comando ↑.

Estando ya bajo estas condiciones, cabe entonces la posibilidad de indicar un proceso continuo, **C** (idóneo en la pantalla y generalmente caótico en la impresora), o no. También somos libres de elección en cuanto al origen de lo representado (**A** fichero auxiliar, **E** ficheros enlazados o ningún valor para el texto en memoria), y en

cuanto al periférico de salida, monitor (**P**) o impresora (**I**).

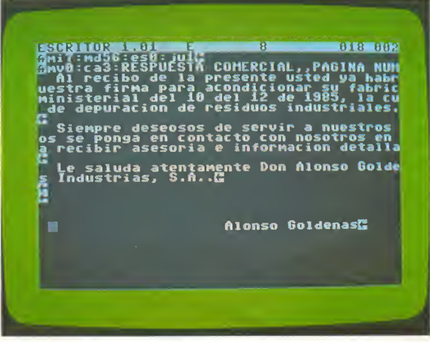
Antes de imprimir, conviene examinar los textos sobre la pantalla y no desperdiciar papel inútilmente. Con este sistema deslizaremos el texto hacia abajo con las teclas Commodore o la barra espaciadora, y pasaremos a la página siguiente con la tecla **C**. El texto se desplazará a izquierda y derecha con las teclas del cursor, el **RETURN, F5** y **F7,** y para finalizar lo indicado es **RUN/STOP.**

ADVERTENCIA: no utilicemos nunca comillas en nuestros escritos, puesto que los caracteres especiales del programa estropean toda la impresión si van entre comillas.

PARA LOS USUARIOS DE LA UNIDAD DE DISCO.

Pues si, pues si, ¿cómo iba a faltar algo parecido en un paquete de software de esta índole? Los afortunados propietarios de una unidad de disco poseen una serie de instrucciones que facilitan su labor, si pulsan **F6.** El potente conjunto de instrucciones abarca desde inicializar una unidad de disco (i), validar un diskette (v) o leer su directorio ($ o $$); hasta formatearlo (n0:nombre,XY, donde XY es un identificativo de dos caracteres), cambiar el nombre de un fichero (r:nuevo nombre=viejo nombre), o borrarlo (s:nombre fichero).





TECNICAS DE FORMATO: UN AS EN LA MANGA DEL PROCESADOR

Formatear un texto significa alterar su curso normal de representación gráfica en pantalla y en impresora, variando desde la forma de las letras (vídeo inverso o caracteres de doble ancho), hasta cambiar el ancho de un documento, crear una cabecera constante en cada página, etc. Todas las sentencias de formato van precedidas siempre por el signo ft (obtenible con **F5**), y no aparecen dentro del texto cuando este se imprime.

Las dos primeras instrucciones a estudiar son miXX (margen izquierdo) y mdXX (margen derecho), cuya función es la de señalar el número de columnas o ancho del texto (mi10: md70, mostrar un texto que empieza en la columna 10 y termina en la 70). Con maXX; línea de texto, crearemos indentados o sangrados, es decir, la línea de texto señalada aparecerá XX espacios más a la izquierda del margen izquierdo indicado. Otro margen asignable es vertical y corresponde al número de líneas en blanco dejadas antes de imprimir (mvXX). El último especimen de este grupo de comandos es una de las capacidades más destacadas del ESCRITOR: la justificación, ju1 alinear el texto a la derecha.

Podemos remarcar el número de caracteres del texto presentes en cada página si antes utilizamos 1pXX (longitud del papel); su va-

lor es igual a la longitud en milímetros dividido entre 4.25. Después, al señalar 1tXX, le diremos al ordenador que incluya XX líneas por página, siempre y cuando 1tXX sea menor de 1pXX.

En cada página podemos incluir una cabecera (caX:aaaa,bbbb,cccc) y un pie de página (ppX:aaaa,bbbb, cccc), y su numeración respectiva al situar el símbolo # en vídeo inverso (**Commodore + G**), en uno de los campos (aaaa,bbbb o cccc), asignado con p=XX, el número de la primera página.

A la hora de crear separaciones entre líneas, esX dejar uno o dos espacios (valores 0 y 1, respectivamente) y InXX mostrar XX líneas en blanco. Posiblemente, queramos añadir comentarios informativos en nuestro documento que no aparezcan en la impresión, utilizando co "comentario".

Por último, la obtención de caracteres en vídeo inverso o doble ancho se realiza mediante los códigos de la impresora, a través de la sentencia: 1=AA, 2=BB, 3=CC, etc., accediendo después a los códigos al pulsar **F1** más el número correspondiente.

EL PROGRAMA ESCRITOR, UNA VERSION MEJORADA DEL EASY SCRIPT PRESENTADO EN CARTUCHO

El procesador de textos de Commodore más representativo ha sido desde siempre el conocido EASY SCRIPT. No obstante, el programa ESCRITOR se asemeja mucho a este (incluso los ficheros son compatibles), aunque anexiona algunas características suplementarias y subsana muchos defectos.

No cabe ninguna duda acerca de la ayuda que nos puede reportar el uso de este paquete de software, especialmente a nivel profesional donde presta servicios múltiples, como las cartas personalizadas y los ficheros auxiliares. El único inconveniente verdaderamente reseñable es su precio.

En resumen, el programa ESCRITOR es un gran procesador de textos para los idiomas castellano, catalán y francés, cuya utilización llegará a ser prácticamente indispensable para muchos de nosotros.



LOS 7 MAGNIFIC

Colaboremos todos en la confección de estos 7 MAGNIFICOS de TU MICRO COMMOROE. Envíanos el nombre de tu programa favorito, dejando muy claro tu nombre y dirección. Todos los meses, sortearemos una suscripción por un año a nuestra revista entre las respuestas recibidas. Anímate y escríbenos a: TU MICRO COMMODORE. (7 MAGNIFICOS). APARTADO DE CORREOS 61.294. 28080 MADRID.



COMMANDO

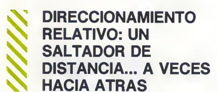
El enemigo es muy superior en número y fuerzas a nosotros, pero no cuentan con nuestro excepcional arrojo y valentía, además de nuestra mayor movilidad al no ser un grupo de fuerzas, sino únicamente un solo hombre. Abandonados a nuestra suerte en medio de las líneas enemigas, para acabar con éxito nuestra misión, consistente en atravesar sus defensas lo más lejos posible, con el objeto de ir abriendo paso a nuestros aliadods. Para ello, habrá que destruir a nuestros oponentes, los cuales nos atacarán desde los lugares más inverosímiles.



...OS
2. RAMBO
3. ONE ON ONE
4. LA LEY DEL OESTE
5. WIZARDRY
6. IMPOSSIBLE MISSION
7. SKYFOX

MODOS DE DIREC

Continuamos explicando cada uno de los modos de nuestra estimada CPU, para señalar a la memoria que desea emplear.



DIRECCIONAMIENTO RELATIVO: UN SALTADOR DE DISTANCIA... A VECES HACIA ATRAS

El modo de direccionamiento relativo se emplea exclusivamente para controlar el flujo del programa. Frecuentemente aparece en las bifurcaciones condicionales (el **IF... THEN GOTO...** del BASIC) y en los bucles.

Abrimos un pequeño paréntesis para indicar que en el código máquina del 6502 (y demás familia), no existe nada similar a los bucles tipo **FOR... NEXT** a los cuales el BASIC nos tiene acostumbrados.

Si queremos ejecutar un bucle, cosa muy frecuente, no queda más remedio que hacerlo en forma explícita, es decir, mientras se verifique cierta condición, volver al punto de partida del bucle.

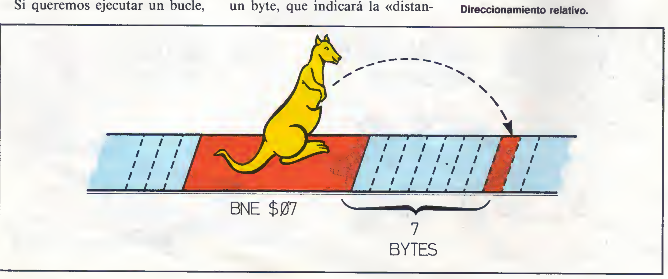
Veremos con más detalle este punto al hablar de los mnemónicos de bifurcación condicional, pero este ejemplo sirve para aclarar el motivo por el cual los programadores evitan el código máquina siempre que pueden: no hay facilidades; todo debemos hacerlo nosotros y esto es difícil, incluso con la ayuda de sofisticados ensambladores o editores.

A continuación del mnemónico de la instrucción (la CPU sabe que instrucción lleva direccionamiento relativo), debemos situar un byte, que indicará la «distancia» que debe saltar el contador de programa. Una instrucción de este tipo podría ser: BNE $10, si se verifica cierta condición salta $10 (16 decimal) bytes.

Para indicar si el «salto» debe realizarse «hacia delante» (es decir, hacia una dirección mayor de memoria) o hacia atrás (hacia una dirección menor), este byte de distancia se codifica de la siguiente forma:

El bit de mayor peso indica el sentido del salto. Si dicho bit se halla a cero (es decir, si el valor del byte es menor de 128), el salto será hacia delante. Si se halla a 1 (valor del byte mayor o igual que 128), el salto será hacia atrás. La instrucción BNE $FA indicará, por tanto, efectuar un salto de 6 bytes hacia atrás.

Direccionamiento relativo.



CIONAMIENTO (II)



EL REGISTRO CONTADOR DE PROGRAMA (PC) TIENE LA CULPA...

Veamos con más profundidad la causa de este embrollo. Como ya sabemos, el registro contador de programa (PC a partir de ahora) contiene en todo instante la dirección de la siguiente instrucción a ejecutar. Si cambiamos el contenido de este registro, cambiamos el flujo del programa, como si hubiésemos ejecutado un **GOTO** del BASIC.

De hecho, esto es lo que hacemos con el direccionamiento absoluto en un programa: se sustituye el contenido de PC por la dirección que indicamos, con lo cual, efectivamente, la CPU «salta» a esa nueva dirección.

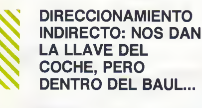
Con el direccionamiento relativo, realmente lo que se hace es sumar (o restar) al contenido de PC el valor del byte que suministramos, con el resultado de hacer «saltar» el PC tantos bytes como le hayamos indicado.

Si efectuamos el ensamblado manual, hemos de tener en cuenta que PC apunta a LA SIGUIENTE INSTRUCCION A EJECUTAR, por lo cual debemos contar a partir de ella a la hora de calcular la distancia de los saltos, tanto hacia atrás como hacia delante. En los listados adjuntos podemos ver un par de ejemplos de esta situación.

```
   :
LDA  0
LDY  2
INX
BNE 2 ──────┐
LDA  10     │
STA$30FG ◄──┘
   :
SALTO HACIA DELANTE...

   :
   LDA  0
   LDX  10
┌─►STA $C000,X
│  INX
└──BNE 250
   :
   O SALTO HACIA ATRAS
```

Los desplazamientos en el direccionamiento relativo, pueden ser, tanto positivos como negativos.



DIRECCIONAMIENTO INDIRECTO: NOS DAN LA LLAVE DEL COCHE, PERO DENTRO DEL BAUL...

El direccionamiento indirecto se basa en un concepto distinto de los modos que hemos visto hasta ahora. En lugar de darnos la dirección del dato, nos indica la dirección donde se halla la dirección del dato.

Es una forma aparentemente enrevesada de entregar información; para darnos la llave del coche, nos entregan la llave del baúl, en cuyo interior encontraremos aquéllas.

¿Por qué esa complicación?. La utilidad de este sistema aparece clara siguiendo con el ejemplo anterior: nuestro hábil amigo puede colocar en el baúl las llaves del Masseratti, del Rolls o del Porsche (si ponemos un ejemplo, pongámoslo a gusto, ¿no?...).

Nuestra agenda (que para nosotros es lo que el programa para el ordenador) nos dice que cojamos las llaves del baúl y salgamos con el coche al que pertenezcan. Y esta es su utilidad: con los distintos contenidos del baúl, tomaremos un coche distinto.

Al ordenador le gustan más los números que los coches, pero el símil es totalmente aplicable: cambiando el contenido de la dirección que indicamos, podemos emplear distintos datos sin modificar el programa.

El 6502 emplea el direccionamiento indirecto para la instrucción de salto incondicional JMP. A continuación del mnemónico, se sitúa entre paréntesis, la dirección donde se halla almacenado el vector (es decir, la dirección de memoria) a la que se desea saltar; por ejemplo, JMP ($C007).

Nótese que la dirección que suministramos es la del primer byte (el menos significativo, recuérdese la inversión) del vector buscado. El segundo byte (el más significativo) se encuentra en la posición siguiente, y esto es algo que la CPU ya supone, por lo cual no hace falta indicación adicional alguna.

Existen otras variantes del direccionamiento indirecto que serán motivo de un próximo artículo.

chispas

CHISPAS

En la posición de memoria 216 del C-64, se encuentra el número de caracteres de inserción pendientes; cuando este número es distinto de 0, los controles de cursor, colores, funciones, etc., no se ejecutan, sino que se escriben los caracteres en reverso tan familiares para nosotros. Pues bien, esto puede servir para ver una variable de cadena (que contenga caracteres especiales), tal como es en realidad. Probemos el ejemplo siguiente en modo directo:

```
A$="{ 2 cursores abajo }
B"
PRINT A$
```

La cadena sitúa una "A" y dos posiciones más abajo y a continuación una "B".

```
POKE216,LEN(A$)
PRINT A$
```

¿Comprendido?

CHISPA

En muchas ocasiones necesitamos poner la variable TI a 0, esto no es posible mediante: TI=0, sino que debe hacerse como si situásemos la hora a cero así: **TI$="000000"** (es decir, 00 horas, 00 minutos y 00 segundos). Sin embargo, con **SYS 65499** conseguiremos el mismo efecto (si, como es lo más normal, los registros A, X e Y están a cero).

CHISPA

Cuando terminan los programas, el ordenador deja un línea en blanco, aparece la palabra READY y el cursor se sitúa debajo de ella; esto a veces hace que el contenido de la pantalla se estropee. Para evitarlo, termina los programas no con un **END** normal, sino con el comando: **SYS 42115.**

CHISPA

Si quieres borrar parte de un programa, de una forma rápida, ejecuta en modo directo POKE 650,128 para tener autorrepetición de teclas, luego mediante un FOR con el STEP adecuado escribe los números de línea a borrar; ahora lleva el cursor sobre el primer número y mantén pulsada la tecla RETURN. Por ejemplo, para borrar las líneas 10-100 numeradas de 10 en 10:

```
FORI=10TO100STEP10:
PRINT I: NEXT
```

CHISPA

Por si no lo sabías, y como curiosidad, el C-64 resuelve tus problemas de trigonometría (SIN, COS, ATN), efectuando un polinomio de quinto grado en el caso de SIN y COS y otro de ¡decimoprimer grado! para ATN. Nuestro C-64 sólo sabe sumar y restar, y a partir de ahí, puede multiplicar (sumas sucesivas), dividir (restas sucesivas) o elevar a potencias (multiplicaciones sucesivas o sumas de sumas sucesivas). ¿Puedes imaginarte lo que «sufrirá» para calcular un ATN?.

CHISPAS

Parece que el teclado del DATASSETTE, es un ente totalmente aislado del ordenador, pues bien, esto no es exactamente cierto, ya que hay un cierto control sobre el mismo. Obviamente, desde el ordenador, no podremos pulsar «PLAY» o «RECORD & PLAY», pero sí podremos detener la ejecución de un programa, hasta que se pulse (por ejemplo), la tecla «PLAY» del DATASSETTE. Esto se conseguirá con la siguiente línea BASIC: Línea IF PEEK(1) =55 THEN Línea.

CHISPAS

El buffer del casete, es una zona controlable mediante punteros; el puntero que se encarga de esta zona en concreto es el 178-179. Así pues, para cambiar el buffer a otra nueva dirección, bastará con variar el puntero anteriormente mencionado. La siguiente línea consigue este efecto:

```
X = Dirección, Nueva:
BA=INT(X/256):BB=X--
BA*256:POKE178,BB:PO-
KE179,BA.
```

CHISPAS

Para borrar una línea de la pantalla, ya hay confeccionada una rutina de la ROM; esta rutina, se puede emplear también para borrar un determinado número de caracteres dentro de la línea. Para ello, situaremos en el registro X el número de línea y en el registro Y el número de caracteres a borrar (comenzando por el cero).

BORRAR LINEA: POKE 781, línea:SYS59903
BORRAR CARACTERES: POKE781, línea: POKE 782, número de caracteres: SYS59905

Donde línea puede tomar un valor entre 0 y 24, y número de caracteres entre 0 y 39.

CHISPA

Si tienes a la vez muchos ficheros abiertos y quieres asegurarte de cerrarlos todos, puedes ejecutar **SYS 62255.** Los cierra de inmediato, pero sólo perfectamente los de entrada; para los de salida mejor es continuar empleando **CLOSE.**

CHISPA

Para saber el último número de línea donde se detuvo el programa, puedes utilizar la siguiente instrucción en modo directo: **PRINT PEEK(59)+PEEK(60)*256.**

CHISPA

Empleando los punteros como en el ejemplo anterior, podemos saber en que posición de la RAM de Video y de Color se encuentra el cursor.

```
PRINTPEEK(243)+PEEK
(244)*256 (Color)
PRINTPEEK(209)+PEEK
(210)*256+PEEK(211)
(Video).
```

CHISPA

En la posición 199 se encuentra la bandera de reverso; si esta posición contiene un 0, la escritura es normal, pero si contiene un 1, lo que se escriba a continuación se imprimirá en reverso. Pruébalo con: **POKE199,1:PRINT "TELEVISION".**

CURSO DE BASIC

EVALUADOR DE CADENAS

De modo general, podemos decir que las funciones BASIC operan como «traductores» que, a partir de un argumento suministrado, producen un valor de salida determinado. De esta forma, la estructura general de cualquier función es:

función (argumento)

Como «función» se especifica cualquiera de las múltiples palabras clave de este tipo disponibles en el vocabulario BASIC, y como «argumento» la expresión a evaluar, la cual puede ser tan complicada como se quiera: una constante, variable o expresión compuesta. En cualquier caso, dependiendo de la clase de función elegida, la expresión «argumento» será de tipo numérico o alfanumérico.

Lógicamente, los argumentos especificados para las diferentes funciones deben ser coherentes con la función escogida, produciéndose, en caso contrario, una detención del programa con el correspondiente mensaje de error **TYPE MISMATCH ERROR** (tipo de expresión inadecuado).

Por los resultados que podemos obtener de ellas, se dividen las funciones BASIC en dos grandes grupos, las de cadena de caracteres y las numéricas. Nos ocuparemos, de momento, de las primeras, pasando más adelante al estudio de las funciones de argumento numérico. Las funciones a que nos referiremos son:

STR$, VAL y **LEN**

De ellas, la primera produce como resultado una cadena de caracteres, mientras que las otras dos proporcionan un resultado numérico. Para poder recordar más fácilmente esta circunstancia, basta con advertir que las funciones de resultado alfanumérico llevan como sufijo el símbolo dólar (**$**) y las de resultado numérico no lo llevan, de forma similar a la regla de construcción de los nombres de variables.

Las funciones pueden admitir como argumento cualquier expresión, siempre que sea del tipo adecuado: de cadena, en este caso. Esto quiere decir que es posible adoptar desde la forma más sencilla, una constante, hasta nombres de variables o expresiones más complejas por combinación de las anteriores, con inclusión opcional de operadores matemáticos, o de otras funciones.



Las funciones BASIC operan como traductores que a partir de un argumento suministrado producen un valor de salida determinado.



Las funciones BASIC se dividen en dos grandes grupos, según los resultados obtenidos de ellas: de cadena y numéricas.

LA FUNCION STR$

Esta función permite obtener la forma de cadena de una expresión numérica. En cierto modo, es como convertir una expresión de un tipo al otro, lo cual es de la mayor importancia debido a que algunas funciones, como ya hemos comentado, sólo son accesibles para determinado tipo de variables o expresiones, y gracias a esta circunstancia, pueden serlo para todas ellas, previa conversión de la expresión al tipo adecuado para la función.

La impresión de variables o expresiones numéricas BASIC, puede representar un problema cuando se desea que aparezcan en pantalla en forma de «columna». Esto es debido a que la sentencia **PRINT** imprime las variables, tanto numéricas como alfanuméricas, de izquierda a derecha, mientras que nosotros estamos más acostumbrados a que las expresiones con texto se justifiquen (columnen o ajusten) por la izquierda, y las numéricas por la derecha.

La solución a este problema es sencilla, y se basa en el empleo de la función **STR$** en combinación con **LEN** y **TAB** o **SPC,** como veremos más adelante.

El motivo de calcular el equivalente en expresión **string** de un valor numérico es, precisamente, poder determinar su longitud; puesto que el BASIC no permite conocer la longitud de una variable o expresión numérica de forma directa, sin su conversión previa a expresión de cadena mediante la función **STR$.**

Este hecho nos puede parecer una considerable incomodidad, pero, sin embargo, se soporta en una sólida base informática; concretamente en la forma en que la máquina trata y almacena la información numérica. Actualmente no estamos capacitados para comprender el por qué de ésto, pero más adelante tendremos oportunidad de conocerlo.

LA FUNCION LEN

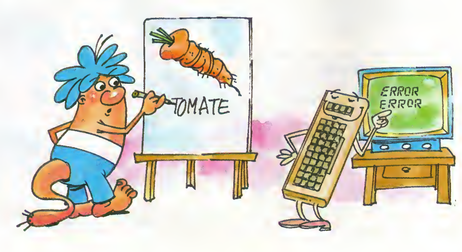
Por medio de la función **LEN** podemos calcular el número de caracteres contenidos en una variable o expresión de cadena. En caso de que deseemos conocer el número de caracteres que ocupará la impresión de una variable o expresión de tipo numérico, deberemos calcular primero la **STR$** de la expresión (**STR$** es abreviatura de **STRing**) para posteriormente hallar su **LEN** (abreviatura de **LENgth** que significa longitud). Veremos más claro todo esto con algunos ejemplos.

```
10 A=-1234
20 B=567890
30 A$="ABCDE"
40 B$="FGHIJKL"
50 PRINT " {CLR  "
60 PRINT "CADENA:"STR$(A),
"LONGITUD:"LEN(STR$(A))
70 PRINT "CADENA:"STR$(B),
"LONGITUD:"LEN(STR$(B))
80 PRINT "CADENA:"STR$(A+B),
"LONGITUD:"LEN(STR$(A+B))
90 PRINT "CADENA:"A$,"LONGI
TUD:"LEN(A$)
100 PRINT "CADENA:"B$,"LONGI
TUD:"LEN(B$)
110 PRINT "CADENA:"A$+B$,"LON
GITUD:"LEN(A$+B$)
```

La función **LEN** permite determinar la longitud (número de caracteres) de que se compone una cadena o una expresión alfanumérica como lo son **A$+B$** o **STR$.**

Hemos visto como incluimos una función dentro de otra con el empleo de paréntesis dentro de paréntesis, en el caso del cálculo de la longitud impresa de una variable numérica. Debemos saber

Los argumentos han de ser coherentes con las funciones escogidas; en caso contrario se producirá un error.





*Las funciones de resultado alfanumérico llevan como sufijo el símbolo dólar (**$**), a diferencia de las de resultado numérico.*

también que el BASIC reserva a la izquierda de cada valor numérico convertido a formato **string** una posición para el signo, la cual permanece en blanco si el valor es positivo y contiene un signo menos en caso contrario. Vamos a comprobar que la **STR$** de un valor numérico, tiene la misma longitud con independencia de su signo:

```
PRINT LEN(STR$("9")),LEN(STR$
("-9"))
```

LA FUNCION VAL

La función **VAL** es la inversa de la **STR$** puesto que calcula el valor numérico de una variable o expresión de cadena. Podemos apreciarlo con algunos ejemplos:

```
10 A=-1234
20 B=567890
30 X$="A+B"
40 PRINT " {CLR} "
50 PRINT X$,VAL(X$)
```

Todos los elementos que se encuentran dentro de una cadena son evaluados para determinar si son o no numéricos, concluyendo la operación al encontrar el primero de ellos que no lo sea, siendo el resultado el valor numérico hallado hasta el momento de la interrupción. Veamos un ejemplo:

```
PRINT VAL("-123456"),VAL("-1
23P456")
```

Una característica de la función **VAL** es la de «normalizar» las expresiones de cadena con el formato estándar de una variable numérica común. Para ello, ignora los espacios en blanco que puedan existir entre los elementos de cualquier expresión numérica, así como los ceros a la izquierda de una cifra.

Esto nos permite efectuar cálculos con variables alfanuméricas que están preparadas para ser impresas sin adaptarse concretamente al formato numérico; por ejemplo, con blancos o ceros a la izquierda, sin que ello produzca ningún error. Por otra parte, podemos emplear la función **VAL** para eliminar este tipo de diferencias respecto al formato estándar:

```
10 PRINT " { CLR } "
20 X$="-   1234":PRINT X$,VAL(X$)
30 X$="-001234":PRINT X$,VAL(X$)
40 X$="0001234":PRINT X$,VAL(X$)
50 X$="-   1234   ":PRINT X$,VAL
   (X$)
60 X$="   1234":PRINT X$,VAL(X$)
```

En este programa, se señalan formas muy diferentes que puede adoptar una cifra y a continuación la normalizada que se consigue con el empleo de **VAL.**

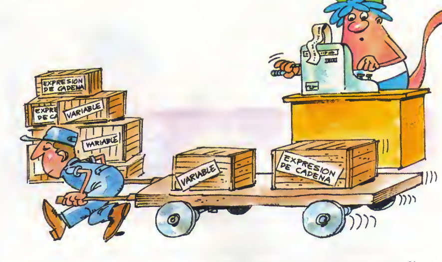
*La función **STR$** permite obtener la forma de cadena de una expresión numérica.*



UN PEQUEÑO REPASO

Con los conocimientos obtenidos hasta el momento estamos capacitados para llevar a cabo un ejercicio de relativa complejidad, basado en el tratamiento de cadenas: realizaremos la toma de diez nombres y los presentaremos en la pantalla separados por cada

Por medio de la función ***LEN****, podemos calcular el número de caracteres que componen una cadena.*



espacio en blanco que encontremos. Por ejemplo, la entrada ANTONIO RODRIGUEZ MENENDEZ será escrita:

ANTONIO
RODRIGUEZ
MENENDEZ

Para separar cada grupo de nombres, dejaremos una línea en blanco. Haremos notar una serie de circunstancias que nos pueden servir de ayuda en la confección del programa. En primer lugar, la estructura principal del mismo, es un bucle que realiza la siguiente tarea: toma el nombre y lo separa para presentarlo tal como se requiere y, finalmente, deja el espacio en blanco para diferenciar el próximo nombre.



El BASIC no permite conocer la longitud de una variable o expresión numérica de forma directa.

Una pista acerca de como localizar los espacios en blanco para separar las partes del nombre entrado, puede ser utilizar un bucle cuyo punto inicial sea 1 (primer elemento de la cadena) y cuyo final sea la **LEN** de la propia cadena. Podemos almacenar en una variable el número de carácter en el que se encuentra el último espacio en blanco detectado, para así tomar la siguiente subcadena desde este punto hasta el próximo blanco:

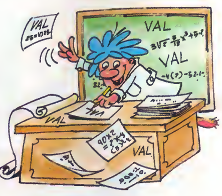
```
10 PRINT " {CLR} "
20 FOR I=O TO 9
30 INPUT"NOMBRE";N$:P=1
40 FOR J=1 TO LEN(N$)
50 IF MID$(N$,J,1)=" "THEN
PRINT MID$(N$,P,J):P=J+1
60 NEXT
70 PRINT
80 NEXT
```

En las líneas 20 a 80 se configura la estructura del bucle basado en la variable **I** que consigue que el bloque comprendido entre las líneas 30 y 70, se repita por diez veces. En la zona de programa comprendida de 30 a 70, se realiza el tratamiento de cada nombre. Este comienza con la entrada del mismo en la variable **N$** mediante un **INPUT** y finaliza con la impresión de la línea de separación para preparar el tratamiento de un nuevo nombre.

Por tanto, la parte más importante de nuestro programa, la que realiza la división del nombre e impresión de sus diversas partes, se encuentra en el bucle anidado de las líneas 40 a 60. Este bucle, controlado por la variable **J,** recorre la cadena de entrada **N$** buscando un espacio en blanco. Si no lo localiza en el carácter estudiado continúa analizando el siguiente, y así sucesivamente, hasta alcanzar el final de la cadena.

Si por el contrario, localiza un espacio en blanco, escribe la parte de cadena analizada hasta el momento y hace que la variable **P** indique el último punto tomado como blanco.



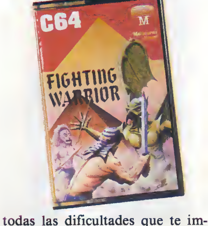


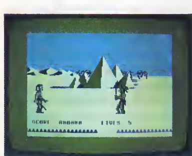
La función ***VAL*** *es la inversa de* ***STR$,*** *puesto que calcula el valor de una expresión o variable de cadena.*

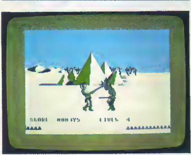
FIGHTING WARRIOR

El misterio nos rodea, puesto que nos encontramos en el antiguo Egipto, el Egipto de los ritos, criaturas místicas y magia oculta. Nuestro amor, la princesa Thaya, ha sido secuestrada por el Faraón; nuestra misión será rescatarla de sus garras.

Escondida en un remoto templo está esperando el momento en que va a ser enterrada viva, como sacrificio a los dioses. Tu, el mejor guerrero del país, armado únicamente con tu valor y una espada, irás en su busca, enfrentándote a grandes peligros y afrontando todas las dificultades que te imponga esa búsqueda.









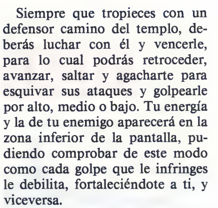
Lucharás contra seres mitológicos, semidioses y aparatos mágicos que los magos del Faraón enviarán contra ti, para impedirte coronar con éxito tu empresa. Además, te encontrarás con los terribles guardas del monarca, avezados luchadores con la espada, preparados para vencer o morir. Por ello, deberás emplear todas tus fuerzas y poder de combate, ya que el mínimo desliz puede suponer tu muerte y por tanto la de tu princesa.

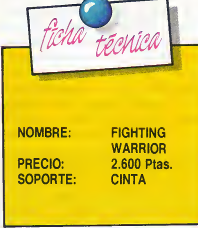
A lo largo de tu odisea encontrarás objetos olvidados por los viajeros en el transcurrir del tiempo, los cuales poseen poderes mágicos, pudiendo teletransportarte a otros lugares; pero ten cuidado porque esos poderes no son fáciles de utilizar y en algunos casos serán malignos y te dañarán.

Si consigues llegar al templo, tu sufrimiento no habrá finalizado, se te pondrán miles de trampas y lucharás contra criaturas aún más poderosas que las anteriores; sólo descansarás cuando encuentres a la princesa y la liberes de su cautiverio.

Siempre que tropieces con un defensor camino del templo, deberás luchar con él y vencerle, para lo cual podrás retroceder, avanzar, saltar y agacharte para esquivar sus ataques y golpearle por alto, medio o bajo. Tu energía y la de tu enemigo aparecerá en la zona inferior de la pantalla, pudiendo comprobar de este modo como cada golpe que le infringes le debilita, fortaleciéndote a ti, y viceversa.

Es posible que cuando venzas a un adversario, te encuentres con un vaso depositado en la arena o el suelo del templo; si lo golpeas, este puede incrementar o reducir tu energía cara a la siguiente lucha, teletransportarte a otro lugar, eliminar la invisibilidad de la princesa o hacerte luchar prematuramente con los dioses enviados por el faraón; todo depende de tu suerte. Así que acumula fuerzas y moral, ¡porque las vas a necesitar!





NOMBRE:	FIGHTING WARRIOR
PRECIO:	2.600 Ptas.
SOPORTE:	CINTA



GRAFICOS:	★★★★★
SONIDO:	★★★★
INTERES:	★★★★
DIFICULTAD:	★★★
originalidad:	★★★★

BREAKDANCE





Ya ha llegado a nuestro país la fiebre del breakdance, y con ella la adaptación de este tipo de danza al ordenador.

En este programa iremos pasando por diferentes fases para demostrar que somos unos perfectos breakers. La primera será el festival de baile de pies calientes, donde seremos observados por todos los bailarines de la ciudad, ya que somos novatos. Estos nos llevarán a una competición en una esquina de la ciudad, donde se encuentra el mejor breaker del barrio, que comenzará su baile, debiendo nosotros imitarle sin perder sus pasos, por lo que tendremos que ser rápidos de cerebro y pies. Cada vez Hot Feet irá realizando su sucesión de pasos más difícil, obligándonos a emplearnos a fondo.

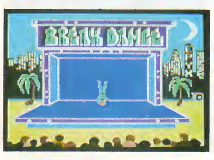
Toda la ciudad ha oido ya hablar de nosotros, y la banda de Rocket Crew ha llegado a nuestra zona, bailando al tiempo que nos empujan hacia el muelle. Será más difícil vencerles ya que deberemos competir con los movimientos de cada breaker, ¡y son muchos!, antes de que consigan tirarnos por el final del muelle.

Como somos ya conocidos por todos, que nos temen por nuestra pericia y rapidez en el baile, nadie se atreve a competir con nosotros, y vamos al encuentro de Boogaloo Brewster, apodado el rey. Este ejecutará una serie de movimientos que deberemos imitar en el mismo orden y recomponerlos a guisa de puzle, claro que gozaremos de un tiempo limitado.

Por fin, después de pasar por numerosas pruebas de las que habremos salido más o menos airosos, hemos sido invitados al festival de break de la ciudad; este es el momento en que deberemos demostrar todas nuestras cualidades. Tendremos un menú de selección en el cual podremos escoger los movimientos que queremos realizar acompañando a la música; estos serán muy variados: Moonwalk, Gyro, Frontflip, Hand Spin y otros muchos, los cuales iremos encadenando en la secuencia que prefiramos para hacer una perfecta coreografía. La velocidad de baile la podremos ir cambiando a medida que ejecutemos los pasos.

Una vez hayamos dominado las cuatro pruebas, podremos acceder al gran Lazo, que nos hará competir secuencialmente en todas las pruebas otorgándonos una determinada puntuación, que puede ser susceptible de entrar en el Hall de la Fama del Breakdance.

En definitiva, Breakdance no es un mal juego, aunque esté basado casi en su totalidad en un perfecto «Simon», que en vez de imitar sonidos, imita pasos de baile, que en el fondo es lo mismo, llegando a hacerse aburrido, por ello sólo merecen la pena sus gráficos, curiosillos al fin y al cabo.







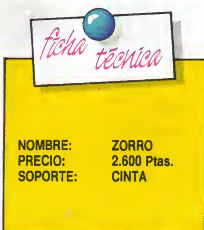
ZORRO

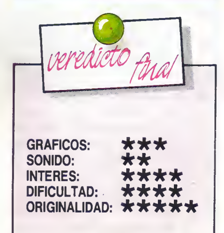
Siempre oimos hablar del Zorro como ese misterioso héroe que luchaba contra la injusticia allá en Méjico; armado con su espada derrotaba a cuantos se ponían a su alcance, rescatando bellas muchachas secuestradas o luchando por los más débiles en la revolución contra los ricos y despiadados jefes de los pueblos. Incluso llegó a ser muy temida su marca personal, con la que humillaba a sus enemigos, la marca del Zorro, que imponía a sus derrotados en la cara con ayuda de la espada, un recuerdo que cicatrizaría y nunca se borraría.

En este caso, el sargento García, militar pendenciero que utiliza su cargo para avasallar a los demás y hacerse rico, ha raptado a una dama y la encarcela en su fortaleza, tu misión es, ya que eres el Zorro, rescatarla de sus garras y darle de paso una buena lección al sargento. Las dificultades serán muy variadas, pues como te puedes figurar, el sargento García te tiene un gran miedo y se ha rodeado de una guardia que defenderá su vida ante cualquier ataque, siempre vigilando toda posible entrada y dispuesta a luchar.

Podrás subir por cuerdas, hiedras adosadas a las paredes, deslizarte sigilosamente por los corredores, saltar de tejado en tejado o caer al suelo desde una altura considerable sin hacerte el menor daño. Por supuesto irás disfrazado con tu atuendo negro que salvaguarda tu verdadera identidad e impide que seas visto con facilidad por la noche, además de propiciar que te deslices como una sombra más por las habitaciones, pasillos y lugares vigilados sin ser



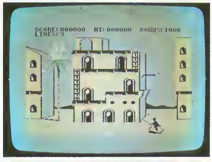


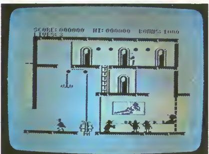


visto fácilmente, y tu rápida espada inflingirá abundantes derrotas en el bando enemigo.

Busca siempre los lugares más propicios para combatir, aquellos donde puedas empujar a un guardia hacia una repisa o tirarle por un tejado, y si quieres derrotarle encuentra el momento justo cuando las espadas estén en posición baja formando una V y asestale el golpe definitivo, dejando por supuesto tu marca personal.

Asimismo, recoge objetos que encuentres, ya que te podrán servir posteriormente, aunque sólo podrás llevar uno cada vez; cuando llegues a una nueva pantalla estudia con tranquilidad las posibilidades de salir de ella, utilizando los objetos que hay en su interior para saltar, tomar impulso o deslizarte por encima de los guardias, las acciones a realizar son mayores de las que parecen a primera vista.





¡FUERA ERRORES!

C-Byte tiene el honor de presentar, en exclusiva nacional para los lectores de TU MICRO COMMODORE, el revolucionario sistema de introducción de programas FUERA ERRORES. Este nos permitirá introducir, sin temor alguno al esfuerzo inútil, cualquier listado por largo y complicado que parezca.

Para adoptar los listados publicados bajo este sistema, deberemos seguir las siguientes normas:

1) Es fundamental transcribir EXACTAMENTE el listado reproducido, incluyendo todos sus espacios, aunque se trate de separaciones entre número de instrucción y línea de instrucción.

2) Todas las líneas finalizarán con un número de tres dígitos, encerrado entre guiones, que NO deberá ser introducido, puesto que no forma parte del programa, sino que tiene la finalidad de hacer funcionar el sistema FUERA ERRORES, según veremos más adelante. Para evitar equivocaciones, dicha cifra entre guiones se sitúa en el margen derecho del final de la línea BASIC a la cual corresponde, a una distancia prudencial del mismo.

3) Para facilitar la introducción de símbolos difícilmente interpretables, se procede a la siguiente representación en los listados.

— Las letras aparecidas entre menor y mayor deberán ser introducidas con pulsación simultánea de la tecla COMMODORE y la letra representada. Ej.: <M> =COMMODORE M.

— Las letras aparecidas entre barras verticales deberán ser introducidas como pulsación simultánea de la tecla SHIFT y la letra representada. Ej.: ¦ K¦ = SHIFT K.

— Entre corchetes simples se representarán los símbolos que se obtienen por pulsación directa de la tecla, aunque lógicamente, este caso sólo se dará para indicar las sucesiones de más de una letra. Así por ejemplo, la introducción de 5 asteriscos se representaría por [5*].

— Para la repetición de símbolos obtenidos mediante las teclas COMMODORE o SHIFT, se seguirá una combinación de las tres normas anteriormente citadas. Así por ejemplo, la introducción de 10 símbolos COMMODORE H, se representaría por [<10 H>].

— Para evitar confusiones, cuando se utilice el sistema de representación de sucesiones de carácter, y éste sea un espacio, se utilizará la abreviatura ESP. [15 ESP] = 15 espacios.

— Los carácteres de control, tales como desplazamientos del cursor, colores, estados de reversa y funciones, se simbolizarán por una abreviatura de tres letras (dos más un espacio en el caso de las funciones) encerrada entre llaves, tal como se señala en la tabla adjunta.

Para introducir cualquier listado por el sistema FUERA ERRORES, deberemos entrar previamente y ejecutar el listado BASIC que aparece en la página siguiente por lo cual es recomendable conservar una copia grabada del mismo, para sucesivas ocasiones. Una vez introducido este listado, ya sea por el teclado, o a través de cinta o disco, debemos ejecutarlo con **RUN.** Instantes más tarde aparecerá en la pantalla el mensaje **FUERA ERRORES!** y el cursor libre para la entrada de programas, con el tradicional **READY.** por encima, lo cual indicará la activación del sistema de depuración de errores.

En virtud al **NEW** que finaliza la línea 20 del programa ¡FUERA ERRORES!, este habrá desaparecido de la memoria, y seremos libres para introducir cualquiera de los programas listados en la sección **TECLA A TECLA** de cualquier número de nuestra revista, o incluso aquellos que apareciendo en otras secciones se acojan a este sistema. Así pues, si el programa ¡FUERA ERRORES! ha desaparecido de la memoria, ¿qué hemos conseguido ejecutándolo? Bien, la respuesta se llama informáticamente **INTERRUPCION:** se trata de una técnica de programación en código máquina que permite que el ordenador efectúe prácticamente dos trabajos a un tiempo, o más correctamente, que ejecute determinada tarea de forma automática, sin necesidad de que le prestemos una atención constante, de forma similar al proceso de respiración en un humano.

Efectivamente, aunque el soporte BASIC ha desaparecido de la memoria, antes de «marcharse» ha dejado funcionando en modo interrupción la pequeña rutina en código máquina que se hallaba en sus **DATAS.** Para comprobarlo pulsa **RETURN**; observarás algo muy extraño: tu ordenador no se comporta normalmente, no sólo desciende una línea el cursor, sino que además hace aparecer un número en la esquina superior izquierda de la pantalla. A continuación veremos como emplearlo.

Cada vez que pulsemos **RETURN**, aparecerá un número en la mencionada zona de la pantalla, y este corresponderá con la instrucción que hayamos introducido. Esto forma parte del sistema de FUERA ERRORES. Cuando introduzcamos cualquier línea de un listado de este tipo, deberemos fijarnos en el número que aparece al pulsar el **RETURN** de fin de línea; si este coincide con el que aprece en el listado al final de la línea, ésta habrá sido introducida correctamente, en caso contrario existe algún error de teclado que debemos modificar. Para modificar una instrucción errónea, no tenemos ni tan siquiera que volver a teclearla si no queremos, bastará sencillamente con modificar el carácter o caracteres erróneos como siempre hacemos, hasta que nos coincida el número de verificación que se presentará al pulsar **RETURN.**

Así pues, el sistema FUERA ERRORES se compone de dos partes: una codificación especial de los listados que facilitan su introducción, evitando los errores al confundir los caracteres gráficos, de control, etc... y un sistema de verificación de líneas que nos advierte en el preciso instante de introducir una de estas, que está mal tecleada.

Ahora bien, el empleo de estos dos sistemas no quiere decir que nos encontremos ante un BASIC diferente al de COMMODORE 64. Este no ha cambiado, hace exactamente las mismas cosas de siempre; simplemente hemos cambiado la forma de hacer los listados. En cuanto al misterioso número que aparece en la esquina de la pantalla no es más que

ERRORES!

una simple suma de comprobación, lo que se conoce técnicamente como un CHECKSUM. La rutina en código máquina de interrupción suma los valores de los caracteres que entramos en cada línea que introducimos, aplicándoles un módulo 256, es decir, volviendo a cero cada vez que su suma parcial supera el 255, de este modo se crea un número entre 0 y 255 dependiente directamente de los caracteres introducidos. Así pues, es prácticamente imposible que una sucesión de errores den por casualidad ese número, mientras que siempre que la línea esté correctamente introducida obtendremos el mismo código que nosotros hallamos por igual sistema y añadimos al final de cada línea al realizar el listado del programa.

Por tanto, el misterioso número no es tampoco una modificación del BASIC de COMMODORE, sino simplemente un pequeño truco para la comprobación de que las líneas han sido bien introducidas. Propiamente no nos evita cometer errores de tecleado, sino que simplemente nos advierte inmediatamente en que línea los hemos introducido.

Para desactivar el sistema sólo deberemos pulsar RUN/STOP RESTORE, y si por cualquier motivo nos interesara reactivarlo, podríamos ejecutar SYS 822, siempre y cuando se encuentre el código máquina en la memoria, lógicamente.

¡ADVERTENCIA! Puesto que el código máquina se encuentra ubicado en el buffer del casete, es imprescindible desactivarlo (RUN/STOP RESTORE) antes de realizar cualquier operación con dicho periférico.

UTILIZACION DEL CASETE CON FUERA ERRORES

Si queremos introducir parte de un programa, para continuar posteriormente el trabajo emprendido, sin perder por supuesto la enorme ventaja del FUERA ERRORES deberemos proceder de la siguiente forma:

1. Desactivar el sistema FUERA ERRORES mediante la pulsación de RUN/STOP + RESTORE.

2. Trasladar el código máquina desde su ubicación en el buffer del casete hasta otro punto desocupado en la memoria RAM (por ejemplo 49152). Para ello sólo hemos de saber que ocupa 114 bytes desde 822 (inclusive). Una sencilla rutina que efectúe el trabajo de reubicación puede ser: FORI=0TO113:POKE49152+I,PEEK(822+I):NEXT.

3. Realizar la operación correspondiente con el casete; ya sea grabar una copia de seguridad de lo introducido en la memoria hasta el momento, ya sea cargar del casete una parte ya grabada del programa.

4. Restablecer FUERA ERRORES a su punto original. Siguiendo el caso de la línea de ejemplo anterior: FORI=0TO113:POKE822+I,PEEK(49152+I):NEXT.

5. Reactivar el sistema FUERA ERRORES mediante SYS 822 (esta vez no aparecerá mensaje alguno, aunque al pulsar **RETURN** comprobaremos que aparecen las cifras de control en la esquina superior izquierda). Lógicamente, los pasos 4 y 5 no son necesarios si después de una grabación se va a apagar el ordenador, y sólo se llevarán a cabo después de las cargas, o si tras la grabación de una copia de seguridad del programa, deseamos continuar introduciéndolo acto seguido.

TABLA DE INTERPRETACION DE CODIGOS DE CONTROL

ABR	SIFNIFICADO	OBTENCION
HOM	HOME	CLR/HOME
CLR	CLEAR + HOME	SHIFT CLR/HOME
ABJ	CURSOR ABAJO	CRSR VERTICAL
ARB	CURSOR ARRIBA	SHIFT CRSR VERTICAL
DCH	CURSOR DERECHA	CRSR HORIZONTAL
IZQ	CURSOR IZQUIERDA	SHIFT CSRS HORIZONTAL
RON	REVERSE ON	CTRL 9
ROF	REVERSE OFF	CTRL 0
F1	FUNCION 1	F1
F2	FUNCION 2	SHIFT F1
F3	FUNCION 3	F3
F4	FUNCION 4	SHIFT F3
F5	FUNCION 5	F5
F6	FUNCION 6	SHIFT F5
F7	FUNCION 7	F7

ABR	SIFNIFICADO	OBTENCION
F8	FUNCION 8	SHIFT F7
BLK	BLACK (NEGRO)	CTRL 1
WHT	WHITE (BLANCO)	CTRL 2
RED	RED (ROJO)	CTRL 3
CYN	CYAN (CIAN)	CTRL 4
PUR	PURPPLE (PURPURA)	CTRL 5
GRN	GREEN (VERDE)	CTRL 6
BLU	BLUE (AZUL)	CTRL 7
YEL	YELLOW (AMARILLO)	CTRL 8
NRJ	NARANJA	COMMODORE 1
MRN	MARRON	COMMODORE 2
RCL	ROJO CLARO	COMMODORE 3
GR1	GRIS 1	COMMODORE 4
GR2	GRIS 2	COMMODORE 5
VCL	VERDE CLARO	COMMODORE 6
ACL	AZUL CLARO	COMMODORE 7
GR3	GRIS 3	COMMODORE 8

```
10 FORI=822TO935:READA:C=C+A:POKEI,A:NEXT
20 IFC<>15254THENPRINTCHR$(147)"ATENCION!, HAY UN ERROR EN LOS DATOS":END
30 PRINTCHR$(147)TAB(213)"FUERA ERRORES!":SYS822:NEW
100 DATA 169,3,141,37,3,169,69,141,36,3,169,0,133,254,96,32,87,241,133,251
110 DATA 134,252,132,253,8,201,13,240,13,24,101,254,133,254,165,251,166,252
120 DATA 164,253,40,96,169,13,32,210,255,165,214,141,176,3,206,176,3,169,0
130 DATA 133,216,169,18,32,210,255,169,19,32,210,255,169,45,32,210,255,166
140 DATA 254,224,100,176,5,169,48,32,210,255,224,10,176,5,169,48,32,210,255
150 DATA 169,0,133,254,32,205,189,169,45,32,210,255,173,176,3,133,214,76,88,3
```

ACOPLAMIENTO

Si te gusta la aventura espacial, ahora puedes pilotar tu propio módulo lunar y despegar de diferentes planetas rumbo a tu nave nodriza.

El programa que presentamos esta semana (aquí el programa..., aquí un lector), puede calificarse más una simulación que un juego. Nuestra misión consiste en despegar de la pista de la base lunar y acoplarnos a la plataforma de la nave nodriza, que se mueve lenta pero majestuosamente (??) sobre nosotros.

Para ello, debemos actuar sobre la barra espaciadora, con golpes secos, continuados pero contundentes, es decir, como si se tratase de nuestra suegra, la cual activara el cohete que impulsa nuestro módulo (la suegra no, la barra espaciadora). La potencia del módulo es un factor que se elige al iniciar el vuelo, de uno a cinco: uno para la potencia mínima y cinco para la máxima.







Pero, como siempre, hay una ley que no podemos violar: la ley de la gravedad, que en su segunda acepción se enuncia como «cuanto más graves estemos, más posibilidades tenemos de morirnos». El factor de gravedad se elige también al comienzo del vuelo, en una escala de uno a cinco. Un factor de dos simula aproximadamente la gravedad terrestre, mientras el máximo podría corresponder a la gravedad de un mundo como Júpiter.

Combinando los factores de gravedad y empuje podemos ver el comportamiento del módulo en distintas condiciones; una gravedad de tipo lunar (factor 1) con un empuje alto harán del vuelo una delicia, mientras que, con gravedad máxima y empuje mínimo ni siquiera será posible el despegue.



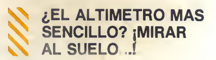
Y LA CRISIS DEL COMBUSTIBLE LLEGA HASTA EL ESPACIO SIDERAL

Como un elemento más de juego, se ha previsto también el escoger la cantidad de combustible que llevamos a bordo (previo pago), en una escala de cero a nueve (mínimo a máximo).

Durante el juego, la cantidad de combustible que queda se indica en la parte inferior de la pantalla, por una barra cuya longitud disminuye a medida que se consume éste. Si el combustible cae por debajo de un cierto valor, sonará una alarma de aviso, y si se acaba... ¡PUFF!

La velocidad de nuestra nave se indica constantemente en el recuadro de la parte inferior derecha de la pantalla. Hay que señalar que las velocidades negativas son de caída, y las positivas de subida.

Tengamos en cuenta que acoplarse con una nave espacial es algo que no puede hacerse «a saco». En la fase de aproximación final hemos de reducir la velocidad al mínimo indispensable, o nos arriesgaremos a chocar con la nave nodriza y a desintegrarnos en una gran explosión, regocijo de los envidiosos que observaban pasmados nuestra habilidad.



¿EL ALTIMETRO MAS SENCILLO? ¡MIRAR AL SUELO...!

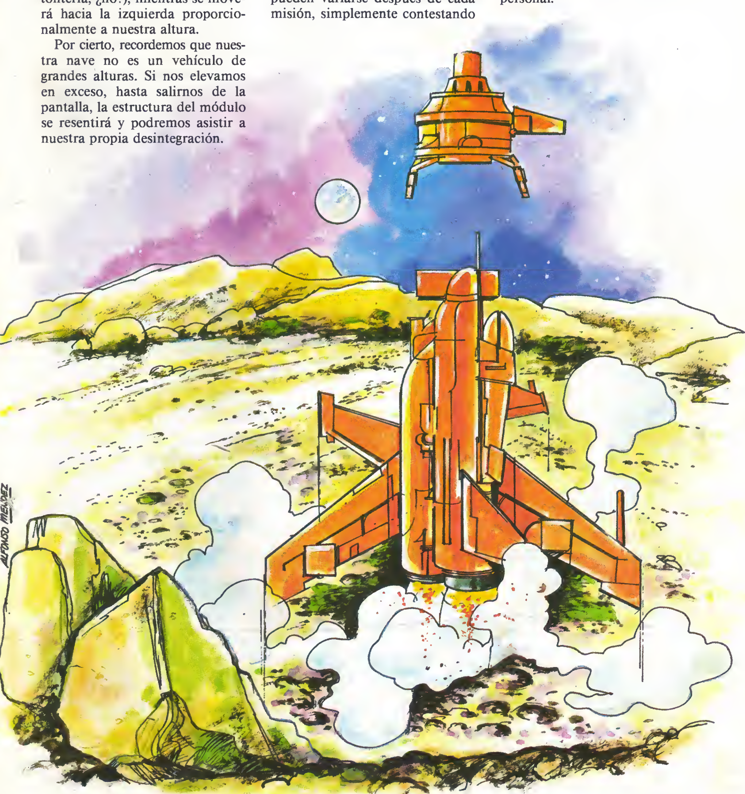
Para controlar la altura de nuestro aparato, no contamos con ningún sofisticado instrumento de la era espacial.

Afortunadamente, la sombra que proyecta nuestro vehículo sobre la superficie puede indicarnos con suficiente aproximación nuestra altura sobre la misma. Cuando la sombra se halle debajo del módulo, estaremos en el suelo (que tontería, ¿no?), mientras se moverá hacia la izquierda proporcionalmente a nuestra altura.

Por cierto, recordemos que nuestra nave no es un vehículo de grandes alturas. Si nos elevamos en exceso, hasta salirnos de la pantalla, la estructura del módulo se resentirá y podremos asistir a nuestra propia desintegración.

En caso de que veamos las cosas muy negras, o simplemente deseemos variar las condiciones de gravedad, empuje y combustible, podemos abandonar el vuelo mediante la pulsación de la tecla "F1". Las condiciones de vuelo pueden variarse después de cada misión, simplemente contestando "s" a la pregunta de variar parámetros.

Que disfrutéis tecleando el listado (eso está claro que es una broma). ¡Ah! Procurad no arruinar a la NASA, que desde lo del Challenger está fatal... de fondos y de personal.



```
10 REM ***********************************          *
                                *                      -217-
15 REM * ACOP II (C) 1986 SINDIN SOFT *                *
                                *                      -244-
20 REM ***********************************             -198-
25 :                                                   -193-
30 POKE56,60:CLR:PRINT"{CLR}":GOSUB375:GOSUB385
                                                       -027-
35 V=53248:P=197:S=54272:C=21:GOSU8435                 -202-
40 CO=FU:A=1:VN=2:8=2:VE=.:H=200:XN=60:GOTO485
                                                       -055-
45 D=(PEEK(P)=60)                                      -196-
50 VE=VE+(PEEK(V+1)<200)*G/100-A*D/E                   -235-
55 IFABS(VE)>5THENVE=5+10*(VE<.)                       -150-
60 :XN=XN+VN:H=H-VE:IFH<30THENH=30:GOTO165             -189-
65 IFH>200THENH=200                                    -120-
70 ON-(XN<245ANDXN>60)GOTO80:VN=-VN:POKEV+9,PEEK(V+9)
+B                                                     -181-
75 IFPEEK(V+9)>70ORPEEK(V+9)<56THEN8=-B                -133-
80 CO=CO-D*(CO>.):IFCO=.THENA=.                        -116-
85 POKEV+21.PEEK(V+21)AND247                           -150-
90 POKEV+1.H:POKEV+8,XN:POKEV+2,-(160-1.5*(200-H))*(H
>95)                                                   -225-
95 IFDTHENPOKEV+7,H+C:POKEV+21,PEEK(V+21)OR(8*A):POKE
V+42,14/(PEEK(V+42)AND7)                               -137-
100 PRINT"{HOM}{23 ABJ}{5 IZQ}{RON}{GRN} ";            -184-
105 A$=RIGHT$("0"+MID$(STR$(INT(100*VE)),2,3),3):IFVE
<.THENA$="{IZO}-"+A$                                   -146-
110 PRINTA$                                            -164-
115 IFPEEK(V+1)=200ANDABS(VE)>.1THEN165                -058-
120 POKES+4,135*A*(-D)                                 -176-
125 IF(PEEK(V+30)AND17)=17THEN150                      -151-
130 POKE1988+36*CO/FU.32                               -104-
135 IFPEEK(P)=4THENPOKEV+21,.:GQTO265                  -043-
140 IF36*CO/FU<6THENPOKES+18,.:POKES+18,33             -038-
145 GOTO45                                             -092-
150 IFPEEK(V+1)<PEEK(V+9)+19THEN165                    -057-
155 IFXN>150ANDXN<160THEN210                           -210-
160 GOTO45                                             -089-
165 POKES+4,.:POKES+11,.:POKES+18,.                    -068-
170 POKEV+21,PEEK(V+21)AND244:POKES+4,.:POKES+5,31:PO
KES+6,1:POKES+1,5                                      -210-
175 POKES+7,128:POKES+8,7:POKES+12,31:POKES+13,1:POKE
S+4,131:POKES+11,17                                    -037-
180 FORI=1TO100:IFI>30THENPOKEV+9,PEEK(V+9)+(PEEK(V+9
)>.)                                                   -202-
185 IFI<30THENPOKEV+21,PEEK(V+21)OR1                   -207-
190 FORJ=1TO90:NEXT:IFI<30THENPOKEV+21.PEEK(V+21)AND2
54                                                     -197-
195 NEXT:POKES+4,.:POKES+11,.:POKES+18,.:PRINT"{ROF}{
CLR}{WHT}{10 ABJ}";                                    -218-
200 A$="[4 ESP]MODULO DE TRANSPORTE DESTRUIDO.":GOSU8
275:PRINT:PRINT                                        -170-
205 FORI=1TO1000:NEXT:GOTO245                          -064-
210 POKES+4,.:POKEV+21,PEEK(V+21)AND247:J=.:POKES+7,1
31:POKES+8.3:POKES+18,.                                -039-
215 POKES+12,131:POKES+13,15:FORI=1TO500:NEXT
                                                       -239-
220 FORI=PEEK(V+9)TO.STEP-1:POKEV+9.I:POKEV+1,PEEK(V+
1)+(PEEK(V+1)>.)                                       -211-
225 POKES+1,J:J=J+1:POKES+4,135:POKES+11,17:NEXT
                                                       -170-
230 POKES+4,.:POKES+11,.:FORI=1TO1000:NEXT             -072-
235 PRINT"{CLR}{WHT}{ROF}{11 ABJ}";:A$="[10 ESP]ACOPL
AMIENTO CORRECTO[9 ESP]"                               -040-
240 GOSUB275:PRINT:PRINT                               -098-
245 A$="[3 ESP]DESEAS MODIFICAR PARAMETROS (S/N) ?[2
ESP]":GOSUB275:PRINT:PRINT                             -072-
250 GETA$:IFA$<>"S"ANDA$<>"N"THEN250                   -069-
255 IFA$="S"THENGOSUB435                               -207-
260 GOTO40                                             -085-
265 PRINT"{CLR}{13 ABJ}":A$="[10 ESP]MISION[2 ESP]ABO
RTADA![11 ESP]"                                        -044-
270 GOSUB275:PRINT:PRINT:GOTO245                       -115-
275 FORI=1TOLEN(A$):PRINT"<+>{IZO}";:FORJ=1TO50:NEXT
                                                       -119-
280 PRINT" {IZO}";:FORJ=1TO50:NEXT:PRINTMID$(A$,I,1);
:NEXT                                                  -222-
285 RETURN                                             -159-
290 END                                                -146-
295 DATA0,24,0,0,255,0,0,24,0,0,36,0,0,90,0,0
                                                       -184-
300 DATA153,0,3,102,192,7,255,224,4,36,32,4,36,32,7,2
55                                                     -206-
305 DATA224,3,254,192,1,231,128,1,63,128,7,231,224,15
,255,240                                               -252-
310 DATA25,195,152,50,60,76,52,126,44,120,255,30,252,
0,63,0                                                 -148-
315 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                -103-
320 DATA0,0,0,0,0,40,170,128,84,85,64,170,168,0,85,80
                                                       -105-
325 DATA0,106,170,0,53,85,0,42,170,128,21,85,64,26,17
0,128                                                  -106-
330 DATA5,85,0,10,42,170,5,149,85,0,0,0,0,0,0,0
                                                       -037-
335 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                -105-
340 DATA0,0,255,192,0,31,224,0,240,192,0,15,0,32,30,2
4                                                      -127-
345 DATA40,60,36,16,124,195,44,131,255,192,255,255,24
8,241,24,143                                           -218-
350 DATA31,255,255,4,66,32,2,66,64,1,36,128,0,255,0,0
                                                       -112-
355 DATA0,28,0,0,59,0,0,117,128,0,234,192,0,117,192,0
                                                       -097-
360 DATA107,128,0,117,128,0,235,128,0,247,128,0,123,1
28,0,63                                                -197-
365 DATA0,0,62,0,0,30,0,0,14,0,0,6,0,0,4,0             -022-
370 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                -104-
375 POKE53280,.:POKE53281,.:FORI=16128TO16383:READJ:P
OKEI,J:NEXT                                            -162-
380 FORI=16064TOI+63:POKEI,255:NEXT:RETURN             -182-
385 PRINT"{CLR}{GRN}{4 ABJ}";                          -193-
390 PRINT" !U!<R>!2 *!!I! !U!<R>!2 *!!I! !U!<R>!2 *!!
I! !U!<R>!2 *!!I![2 ESP]!U!!*!<R>!*!!I! !U!!*!<R>!*!!
I!"                                                    -077-
395 PRINT" !2 -![2 ESP]!-! !2 -![2 ESP]!-! !2 -![2 ES
P]!-! !2 -![2 ESP]!-![4 ESP]!-![5 ESP]!-![2 ESP]"
                                                       -136-
400 PRINT" !2 -![2 ESP]!-! !2 -![2 ESP]<E> !2 -![2 ES
P]!-! !2 -![2 ESP]!-![4 ESP]!-![5 ESP]!-![2 ESP]"
                                                       -079-
405 PRINT" !-!<O>!2 *!<W> !2 -![4 ESP]!2 -![2 ESP]!-!
 !-!<O>!2 *!!K![4 ESP]!-![5 ESP]!-![2 ESP]"            -163-
410 PRINT" !2 -![2 ESP]!-! !2 -![2 ESP]<R> !2 -![2 ES
P]!-! !2 -![7 ESP]!-![5 ESP]!-![2 ESP]"                -148-
415 PRINT" !2 -![2 ESP]!-! !2 -![2 ESP]!-! !2 -![2 ES
P]!-! !2 -![7 ESP]!-![5 ESP]!-![2 ESP]"                -196-
420 PRINT" <2 E>[2 ESP]<E> !J!<E>!2 *!!K! !J!<E>!2 *!
!K! <2 E>[5 ESP]!J!!*!<E>!*!!K! !J!!*!<E>!*!!K!"
                                                       -148-
425 PRINT"{2 ABJ}";:A$="[8 ESP](C)[2 ESP]TUMICRO[2 ES
P]COMMODORE.[8 ESP]":GOSUB275                          -107-
430 PRINT"{2 ABJ}";:RETURN                             -255-
435 PRINT"{GRN}";:A$="[6 ESP]GRAVEDAD[5 ESP](1-5) ? "
:GOSU8275:PRINT"{RON} {ROF}{IZO}";                     -006-
440 GETA$:IFA$<"1"ORA$>"5"THEN440                      -096-
445 PRINTA$:PRINT                                      -118-
450 G=4.5*VAL(A$):A$="[6 ESP]EMPUJE MOTOR (1-5) ? ":G
OSUB275:PRINT"{RON} {ROF}{IZO}";                       -112-
455 GETA$:IFA$<"1"ORA$>"5"THEN455                      -108-
460 PRINTA$:PRINT:E=8-VAL(A$)                          -045-
465 A$="[6 ESP]COMBUSTIBLE[2 ESP](0-9) ? ":GOSUB275:P
RINT"{RON} {ROF}{IZO}";                                -067-
470 GETA$:IFA$<"0"ORA$>"9"THEN470                      -105-
475 PRINTA$:PRINT:FU=50+20*(VAL(A$)+1)                 -237-
480 RETURN                                             -156-
485 POKE2040,252:POKE2041,253:POKE2043,255:POKE2044,2
54:POKE2042,251                                        -090-
490 POKEV+39,12:POKEV+40,.:POKEV+41,.:POKEV+42,7:POKE
V+43,12                                                -076-
495 POKEV+4,145:POKEV+5,220:POKEV+27,4                 -254-
500 PRINT"{GRN}{CLR}":POKEV,165:POKEV+1,200:POKEV+6,1
65                                                     -127-
505 POKEV+9,55:POKEV+23,12:POKEV+29,20:POKEV+2,162:PO
KEV+3,203                                              -185-
510 POKES,130:POKES+1,10:POKES+5,131:POKES+6,131:POKE
S+24,15                                                -055-
515 POKES+14,130:POKES+15,10:POKES+19,195:POKES+20,24
0                                                      -142-
520 PRINT"{MRN}{HOM}{12 ABJ}";:                        -113-
525 PRINT"[12 ESP]{RON}<RON>[2 ESP]!ESP!{9 DCH}<DCH>!
DCH!{13 DCH}";                                         -040-
530 PRINT"{11 DCH}<DCH>[5 ESP]!ESP!{2 DCH}<DCH>{3 DCH
}<DCH>[2 ESP]!ESP!{7 DCH}<DCH>!DCH!{3 DCH}";
                                                       -215-
535 PRINT"{10 DCH}<DCH>[7 ESP]!ESP!<ESP> !ESP!{DCH}<D
CH>[4 ESP]!ESP!{5 DCH}<DCH>[2 ESP]!ESP!{2 DCH}";
                                                       -179-
540 PRINT"<3 T><2 Y><T><2 U><Y>!N![8 ESP]<M><@>[2 ESP
]!M![6 ESP]!M! <2 T>!N![4 ESP]!ESP!{DCH}";             -073-
545 PRINT"[6 ESP]!U!!I! !M!<4 P>!N!!M!<2 @>!N! !M!!N!
<Y>[11 ESP]<2 P>[4 ESP]";                              -050-
550 PRINT"[6 ESP]!2 -![25 ESP]!N! !N!<H>[3 ESP]";
                                                       -048-
555 PRINT"[6 ESP]<V><C>[25 ESP]!O!!P! <G>[3 ESP]";
                                                       -233-
560 PRINT"[10 ESP]!Q![3 ESP]!N!<8 T>!M![3 ESP]!W![5 E
SP]<H><N><G>[3 ESP]";                                  -069-
565 PRINT"[3 ESP]!U!!I![4 ESP]!W![3 ESP]!N![3 ESP]!N!
[2 ESP]!M![3 ESP]!M![3 ESP]!Q![4 ESP]!L!<L>!N![4 ESP]
";                                                     -060-
570 PRINT"[2 ESP]!N!!J!!K!!M![2 ESP]!Q![3 ESP]!N![3 E
SP]!N!<4 Y>!M![3 ESP]!M![3 ESP]!W! <A>!7 *!<S>";
                                                       -112-
575 PRINT"[3 ESP]<N>[3 ESP]!W![3 ESP]!N![14 ESP]!M![3
 ESP]!Q!!-!{GRN}VEL [3 O]{MRN}!-!";                    -254-
580 PRINT"[11 ESP]<16 Y>[4 ESP]<Z>!7 *!<X>{GRN}FUEL{R
ED}[5 ESP]{HOM}";                                      -220-
585 FORI=1988TO2023:POKEI,81:NEXT:POKEV+21,31:GOTO45
                                                       -236-
```

el tablón

TU MICRO COMMODORE
(EL TABLON)
APARTADO de CORREOS 61.294
28080 MADRID

Os recordamos nuevamente, que ninguna carta con ánimo de lucro o que huela tan sólo ligeramente a «pirata», olor fétido de descomposición cerebral fácilmente reconocible, será publicada en nuestras páginas.

Además, aprovechamos la ocasión para haceros un ruego: escribid el texto de vuestros anuncios en un papel a parte de cualquier otro que enviéis con sugerencias, etc..., y emplead letra de imprenta lo más clara que podáis, a máquina siempre que os sea posible.

Vendo C-64 datassette C2N y joystick con seis meses de uso por 45.000 pesetas. Con la compra regalo 100 juegos y utilidades como Koala Pad, Pitfall II, One on one, Sintetizador de voz, etc. También vendería sólo el C-64 por 40.000 pesetas. Llamar de lunes a jueves, a partir de las 16 horas, al tel.: (93) 223 13 57, preguntar por Enrique Perlado Casas. C/ Rabi Ruben 6B, ático. 08004 Barcelona.

Vendo videojuegos T.R.Q.: Consola 10.000 pesetas, seis cartuchos 5.000 pesetas. Todo por 10.000 pesetas. Dirigirse a C. Urtasun. García Ximénez, 5, 1 izquierda. 31002 Pamplona.

Vendo C-64, datassette, joystick, programa recibos, programa órgano, pipeline, falcon patrol, skramble, ajedrez, donkey, kong y otros, libros, revistas, todo 50.000. Ismael Navarro Ballesteros. Cami Can Comelles, 2, 7.º, 1.ª. Esparraguera. Barcelona.

Cambio SPECTRUM 48k (con manuales, cables, más 300 juegos, interface Kempston, 90 revistas) por unidad de disco para CBM-64. Mi dirección: Mario Castells. Valencia, 17. Tel.: (93) 432 19 43. 0815 Barcelona.

Deseo intercambiar programas, tengo unos 180, entre ellos Ghostbusters, Pit stopII, Logo, etc., en cinta, enviar lista a: Andrés Ortega. Fidel Sarel, 13, 1.º izqu., 1.ª. 50013 Zaragoza.

Me gustaría comprar ordenador C-64 por menos de 25.000 pesetas. Mandadme lista. Joaquín Roca i Mallarach. Clot Rosat, 3. Olot. Gerona. Tel.: 26 03 06.

riencias; pretendemos utilizar las amplias posibilidades del C-64 para investigar sobre música, gráficos, y computer-art. Nuestro boletín interno será una revista por ordenador. Interesados escribid a C.I.S. Apdo. correos 2886, BARCELONA 08080.

Vendo ordenador VIC-20 con unidad de cinta por 25.000 ptas. Interesados llamar a (93) 630 00 07. Preguntar por Angel Cabello.

Desearía cambiar programas tanto de C-64, como C-128. En floppy o en casete. Mi dirección es: Salvador Pou. Paseo del doctor Moragás, 204. Barbera del Vallés. BARCELONA.

Atención Commodorianos. Desearía intercambiar todo tipo de juegos y de utilidades para C-64. Poseo los mejores juegos: Rambo, Entombed, Skyfox, Frank Bruno's boxing, Exploding Fist, Dambusters. Prometo contestar. Joan Carles Paris. P. Verdaguer, 138, 1.º 2.ª. Igualada. BARCELONA. Tf.: 863 52 56.

Vendo Commodore 16 como nuevo (ENERO 85), casete, cables, manuales. El mejor BASIC del mercado (do, loop, until, while, circle, else, etc) dos cintas originales, una de ellas la más vendida en UK (BMX RACERS), lote de revistas de Commodore, monitor de código máquina incluido en ROM. Todo por 38.000 Ptas. Narciso Mesa Bravo. Ronda del Salvador, 85. Plasencia, CACERES. Tfno.: 411667.

CBM 64. Intercambio programas. Poseo unos 700. Mandar lista a: Javier Hernández Sole. C/ El Roser, 34. Mora la Nova, TARRAGONA. Tfn.: 401084.

Intercambio programas en cinta para C-64. Interesados enviar lista. Jordi Aymerich. C/ Trebol, 7. Sarriá de Ter, GIRONA. Tfn.: (972) 209755.

Vendo Commodore 64, unidad lectora de casetes, manual de instrucciones, dos tomos de un curso de BASIC, la guía de referencia, muchas revistas, documentos y garantia, más de 90 programas en turbo, juegos aplicaciones, lenguajes: todo a precio interesante. Llamar preguntando por Fernando, al teléfono de BARCELONA (93) 254 92 72.

Desearía vender un INTERFACE, para cualquier cassette compatible para COM. 64 y Vic 20. Comprado desde el mes de Octubre de 1985 hasta principios de enero, se encuentra en excelente estado, vale para cualquier aparato de cassette. Llamar al 404-33-86 a la hora de comer o desde las 9 de la noche en adelante; precio 1.700, preguntar por Raúl.

Vendo impresora Commodore (SEIKOSHA 250X) seminueva por 25.000 ptas. Interesados llamad a horas de comidas al 445 83 35, prefijo 91 para llamadas de fuera de Madrid. Preguntar por José Luis.

Compro ordenadores estropeados. Desde 3.000 hasta 6.000 pesetas. Da lo mismo la marca. Pedro David Casardo Santiago. C/ Padre Claret 50-62 4.º 2.ª escalera derecha BARCELONA. Teléfono 258 77 40. Después de las 9 de la noche.

Club (C) LDA. Intercambios a todo nivel para 64, 128 y Amiga. Extensa biblioteca de soft original y posesión de hard. Para contactar: Andrey Ibáñez i Perales. C/ Passeig de Ronda 76, 3-. (25006), Lleida (para clubs y particulares).

Vendo VIC-20 con fuente de alimentación, unidad de ciclo y pantalla, con libros de programación, todo muy nuevo, sin usar (2 semanas); a buen precio. Harald Schillarg. C/ Cuatro 12, Castelldefels. (Barcelona). Tfno. 93-665 24 33.

¿Quieres participar en un club de estudiantes de B.U.P. que crea sus propios programas para ayudarse en los estudios? Escribidnos a C.E.B. CO-64. C/ Virgen de los Milagros 96, Puerto de Santamaría. (Cádiz).

Tengo un VIC-20. Quisiera establecer contactos con otras personas que también lo tengan, para intercambiar programas. Pedro Simona C/ Portal de Cerdana. Seu de Urgen (Lérida).

Atención oferta cambio por Commodore 64 y Datassette en buen estado, emisora de radio aficionado Maxcon CB 40 canales, móvil o fija, antena magnética para automóvil, ídem. fija, 40 metros cable blindado antena, fuente alimentación, medidor de estacionarias, manual fácil de radio aficionado. Interesados llamar al tel.: (93) 421 76 01, o escribir a Antonio Lázaro. Iberia, 9. 0814 Barcelona.

No soy «pirata», pero por razones económicas vendo programas que ya no utilizo, así como un juego de luces rítmicas (cuatro canales), una emisora de FM (3W) y algunas cosas más (programas de Commodore-64). Tel.: 472 03 99 Madrid. También deseo contactar con programadores amigos del C-64.

Si quereis intercambiar programas, experiencias, trucos, proyectos e iniciativas para C-64, escribir al Club Intersoft. Apartado de Correos 27016-08020 Barcelona; o a Manuel Arroyo, C/ Maresma 266, Barcelona. Entra en el primer Club de intercambio de iniciativas y programas para el C-64.

Vendo VIC-20, ampliación 16 K, tarjeta de carga rápida, libros: guía del usuario, guía de referencia del programador, curso de BASIC en dos tomos. Llamar por la noche al 254 80 62 de Madrid.

Intercambio programas para C64 en cinta; juegos y utilidades (K. Pad Dambusters, Entombed, Superzaxxon,...).
Contesto a todas las cartas. Escribir a Juan Fernando Uso Cervera. P/ Juan Bautista Tejedo, n.º 3. Burriana (Castellón).

Intercambio programas para el CBM-64 por otros de gestión o juegos. Escribid a Jesús Blanco Garcillán. C/ Calasparra, 23. 28033 MADRID.

Desearía intercambiar todo tipo de programas para CBM 64. Manda tú lista, prometo contestar. Santi Espinet Mendiola. C/ Estanislao Figueres, n.º 8, 2.º 1.º. Terrassa BARCELONA.

LISTADOS DESDE PROGRAMA

En la programación de esta semana, vamos a tratar de extraer todo el jugo al comando *LIST*, haciendo listados desde programas, pero sin perder el control del mismo.

En el mundo del lenguaje BASIC existen muchísimos dialectos. Estos dialectos generan nuevas sentencias mejoradas o simplemente cambian la sintaxis de otras ya existentes; aún así, tenemos unas cuantas que son clásicas del BASIC; **FOR, PRINT, NEXT**..., y también **LIST.** Lo que ya no es tan normal es que **LIST** sea un comando programable como es el caso de nuestros ordenadores Commodore.

Normalmente, el comando **LIST** sólo es ejecutable en modo directo, ya que, ¿para qué podría servir este comando en un programa? Bien, aquí debemos dejar volar nuestra imaginación, y seguro que siempre, a poca facilidad que se nos dé, le sacaremos alguna utilidad práctica. En este caso concreto, y pensando un poco, podemos ver que podría ser útil para mostrar ciertos trozos de listado o subrutinas, dentro de un programa de enseñanza; o bien para automodificar alguna parte del listado desde programa, o simplemente para listar o editar un programa en pantalla o impresora más cômodamente.

Pero este comando, tiene algunos inconvenientes que nos impiden trabajar con el de la manera que nos gustaría.

Uno de ellos, es que no se pueden listar zonas variables, por ejemplo, para listar el programa entre las líneas 100 y 200 tendremos que emplear siempre: **LIST 100–200,** y no: **H=100:B=200: LIST a–b.** Por tanto, los extremos del listado han de ser siempre literales, nunca variables.

El segundo inconveniente es el que vamos a tratar de solucionar en esta sección: se trata de la facilidad de efectuar un listado desde dentro de un programa, pero contrarrestada por la pérdida del control del programa. Es decir, ejecutando **LIST** desde programa, pasamos de modo programa a modo directo; por lo tanto, esto sólo sirve para terminar el programa con un listado íntegro o parcial del mismo.

Para no perder el control del programa después del listado, debemos proceder engañando al ordenador, ejecutando lo obvio: abrimos la pantalla como periférico de salida, lo cual se efectúa en la línea 10 del programa ejemplo; a continuación, se mandan dos caracteres al buffer del teclado: un 19 (home) y un 13 (return), para que si se encuentra escrito el mensaje **GOTO 40,** se ejecute en modo directo desde programa, ¿complicado? ¡es muy fácil! Antes de ir a la línea 40, que es la encargada de cerrar el fichero de pantalla y continuar el programa, se habrá efectuado el listado de la parte del programa que nos interese.

Si queremos listar condicionalmente unas partes de programa u otras, se incluirían entre la línea 30 y 40 todas las posibles opciones con **IF**:

IF F=1 THEN LIST 100–190
IF F=2 THEN LIST 200–290
..................etc.

En la línea 30, se sitúa el color de los caracteres igual al de fondo, para que el mensaje **GOTO 40** no estropee la pantalla por motivos de estética; una vez realizado esto, se otorga un color a los caracteres que supera en uno al del color de fondo.

En la línea 40 se escribe el mensaje de continuación, y en la línea 50 se espera la pulsación de una tecla, resuelto lo cual, la continuación del programa está representada por la línea 60, que escribe la palabra «fin» y termina el programa.

```
10 PRINT"{CLR}":OPEN3,3                                     -227-
20 POKE631,19:POKE632,13:POKE198,2                          -222-
30 POKE646,PEEK(53281):PRINT"{CLR}GOTO40":POKE646,PEE
K(53281)+1:CMD3:LIST100-200                                 -087-
40 PRINT#3:CLOSE3:PRINT"CONTINUA EL PROGRAMA, PULSA U
NA TECLA"                                                   -082-
50 POKE198,0:WAIT198,1                                      -032-
60 PRINT"{CLR}FIN":END                                      -216-
100 REM LINEA 100                                           -207-
110 REM LINEA 110                                           -209-
120 REM LINEA 120                                           -211-
130 REM LINEA 130                                           -213-
140 REM LINEA 140                                           -215-
150 REM LINEA 150                                           -217-
160 REM LINEA 160                                           -219-
170 REM LINEA 170                                           -221-
180 REM LINEA 180                                           -223-
190 REM LINEA 190                                           -225-
200 REM LINEA 200                                           -209-
```

MOVIENDO MEMORIA CON RUTINAS ROM

En muchas ocasiones, nos vemos ante la necesidad de trasladar un bloque de memoria de una zona a otra.

Dependiendo del tamaño del bloque a mover, y haciéndolo (incluso en modo directo) desde el BASIC, el tiempo necesario para su ejecución puede ser desde pocos segundos, hasta varios minutos.

Pero no es necesario esperar tanto, podemos hacer una rutina en código máquina que lo haga en menos tiempo; aunque lo mejor es usar las rutinas ya confeccionadas en la ROM de nuestro Commodore. Parémonos por un momento a pensar que debe haber por lógica una rutina de este tipo, ya que al insertar una línea en un programa, todo él debe subir en la memoria el mismo número de bytes que ocupa la línea; o bajar, si borramos otra línea. En efecto, hay varias rutinas disponibles.

Una de las aplicaciones más interesantes de esta rutina, es copiar el juego de caracteres de la ROM a la RAM para modificarlo a nuestro antojo; esto es precisamente lo que hace el programa que presentamos, y que se puede usar como una rutina en nuestros programas.

Como dato curioso, diremos que copiar todos los caracteres del BASIC, en total 4096 (256 × 8 × 2) bytes, puede tomar alrededor de un minuto, mientras que con nuestro programa se efectúa la copia en media décima de segundo, es decir, multiplicamos la velocidad por 400.

Los nombres de las variables empleadas son bastante explícitos; pero para utilizar la rutina con otras direcciones, veamos que variables habrá que tocar:

MOVE: Es la dirección de inicio de la rutina y, por lo tanto, quedará siempre inalterada. Los usuarios del VIC 20 deben cambiar su valor a 50156.

INICIO: Es el valor de la primera posición a trasladar.

FIN: Contiene la dirección del último byte a trasladar.

DESTINO: Dirección final.

NFIN: Nuevo fin del bloque copiado. También calculado por la propia rutina.

A, B, C, A %, B %, C %: Variables auxiliares tipo puntero. No necesitan modificación.

Hay que hacer notar que esta rutina copia perfectamente bloques que no se superpongan. Por ejemplo, de 50000 a 51000 hasta 20000-21000. También funciona bien para copias superpuestas hacia arriba. Por ejemplo, de 4000 a 4100 hasta 4010-4110. Pero no funcionará con bloques superpuestos hacia abajo, como sería: de 4000 a 4100 hasta 3990-4090. En este último caso, se copiará el último byte (4100) en todo el bloque (3990-4090); esto puede ser útil para llenar con un byte una zona de memoria: por ejemplo, llenar de ceros la pantalla de alta resolución (borrar la pantalla).

```
100 REM RUTINA MOVE MEMORY                                -216-
110 REM COPIA JUEGO DE CARACTERES                         -098-
120 POKE52,48:POKE56,48:CLR                               -104-
130 POKE56334,PEEK(56334)AND254                           -253-
140 POKE1,PEEK(1)AND251                                   -083-
150 GOSUB190                                              -208-
160 POKE1,PEEK(1)OR4                                      -191-
170 POKE56334,PEEK(56334)OR1                              -101-
180 POKE53272,(PEEK(53272)AND240)+12:END                  -233-
190 MOVE=41964                                            -054-
200 INICIO=53248                                          -176-
210 FIN=INICIO+4096:FIN=FIN+1                             -019-
220 DESTINO=12288                                         -012-
230 BYTES=FIN-INICIO:NFIN=BYTES+DESTINO                   -168-
240 A%=BYTES/256:A=BYTES-256*A%                           -069-
250 B%=(NFIN-A)/256:B=NFIN-256*B%-A                       -190-
260 C%=(FIN-A)/256:C=FIN-256*C%-A                         -038-
270 POKE781,A%+1:POKE782,A                                -237-
280 POKE91,C%:POKE90,C                                    -040-
290 POKE89,B%:POKE88,B                                    -053-
300 IFA=0THENSYS(MOVE+7):RETURN                           -034-
310 SYS(MOVE):RETURN                                      -085-
```

CONCURSO

El travieso C-Byte tiene el honor de invitaros a la participación en nuestro tercer concurso de programación. Los requisitos necesarios son bien pocos:

- Saber programar un ordenador COMMODORE.
- Ser español o extranjero y
- Tener una edad comprendida entre 5 y 105 años.

Fácil, ¿verdad?
En cuanto a los premios, la mar de atractivos:

- **1.er premio.–60.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **2.° premio.–30.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **3.er premio.–15.000 pesetas en material informático a escoger por el premiado, más un poster de C-Byte.**

Y en fin, si alguno de los programas destaca por su originalidad, estética o comicidad, no sería de extrañar que le cayera alguna cosilla más...

Las bases del concurso son bien sencillas:

❶ Los programas remitidos al concurso deberán ser creación original del autor o autores, y completamente inéditos, pudiendo remitir tantos programas como se desee.

❷ Los programas deberán ser enviados en casete o disco flexible a TU MICRO COMMODORE (Concurso de programación). Apartado de correos 61.294. 28080 MADRID.

❸ Los programas podrán ser de cualquier tipo (juegos, utilidades, gestión, educativos) y habrán de estar escritos en lenguaje BASIC o código máquina.

❹ Los programas deberán ser remitidos desprovistos de cualquier tipo de protección, que impida o dificulte el análisis del mismo, así como reproducción en estas páginas y su introducción como listado siguiendo el sistema FUERA ERRORES.

❺ Cuando la ejecución del programa precise de la concurrencia de determinado periférico o aditamento (joysticks, tabletas gráficas, programas comerciales de ayuda), se valorará la indicación de las modificaciones pertinentes, para que el programa pueda ser disfrutado por cualquier usuario en la configuración básica.

❻ Todo programa presentado al concurso deberá acompañarse de los siguientes datos:

- Datos personales del concursante.
- Nombre del programa.
- Modelo para el que está destinado.
- Breve descripción del programa detallando las indicaciones necesarias para su ejecución.

❼ Los programas premiados pasarán a ser propiedad de la revista TU MICRO COMMODORE, pudiendo hacer ésta libre uso de ellos, y renunciando sus autores a cualquier otra compensación distinta al premio

❽ Los programas no premiados que por su calidad se hagan merecedores de su publicación, serán adquiridos por la editorial, aplicando la tarifa vigente

❾ Los programas recibidos con posterioridad a la fecha tope de la presente edición, serán automáticamente incluidos en los destinados a la siguiente.

❿ El jurado decidirá sobre todos los aspectos no contemplados en estas bases y su decisión será inapelable.

Y ahora a darse mucha prisa, el plazo para la recepción de programas termina el próximo día 15 de mayo (mes de las flores).

¡¡SUERTE!!







La revista semanal donde el usuario de COMMODORE podrá encontrar gran variedad de programas y noticias, sus periféricos y equipos.

SUSCRIPCION POR TELEFONO

Todos los días, excepto sábados y festivos, de **9 a 15,00 y de 17 a 19,00 horas** atenderemos sus consultas en el ☎ 2505820/2505579.
¡Suscríbase por teléfono! y en pocos días tendrá en su domicilio TU MICRO COMMODORE.

SUSCRIPCION POR CORREO

Deseo suscribirme a la revista TU MICRO COMMODORE al precio de 7.400 ptas. **(incluido IVA),** por el período de un año a partir del N.º...... inclusive.

El importe **—que abonaré en su totalidad con el primer envío—** lo haré efectivo de la siguiente forma:

☐ Talón bancario adjunto a nombre de INGELEK, S. A. ☐ Giro Postal n.º ....
☐ Tarjeta VISA n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_| FIRMA
☐ Tarjeta MASTER CARD n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
Fecha de caducidad de la tarjeta .... Nombre titular ....

APELLIDOS |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| NOMBRE |_|_|_|_|_|_|_|_|_|_|_|_|
DOMICILIO |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| EDAD |_|_|
CIUDAD |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| PROVINCIA |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
C. POSTAL |_|_|_|_|_| TELEFONO |_|_|_|_|_|_|_|_| PROFESION |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Recorte o copie esta tarjeta y envíela dentro del sobre a: Ediciones INGELEK Apartado de Correos 61.294. 28080 MADRID

COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.



Más 64's que nadie.

El C-64 de Commodore sigue siendo el ordenador personal más vendido del mundo por sus prestaciones y posibilidades.

Más periféricos que nadie La Gama de periféricos y accesorios del C-64 multiplica sus funciones de una forma casi ilimitada: impresoras, unidades de disco, monitores... Todo un mundo informático a su alrededor para que usted le saque todo el partido.

Más software que nadie El C-64, por ser el ordenador más vendido, ha hecho que las compañías de software se vuelquen en él, creando un parque de programas que hoy le convierten en el 64 con más software del mercado. Software que abarca todos los sectores, desde los negocios hasta el educativo.

Le podemos asegurar que hoy por hoy el programa que usted necesita ya lo tiene el Commodore 64.

Más información que nadie El C-64, lejos de quedarse atrás y porque cada vez son más los que lo eligen, dispone del mayor número de publicaciones exclusivas, así como libros de documentación en castellano donde se tratan temas de interés, nuevos programas, nuevas ideas, nuevas aplicaciones...

Cada vez más gente investiga y se preocupa por su Commodore 64.

Y menos precio que nadie Y todo esto a un precio realmente sorprendente y asequible.

Acérquese hoy mismo a un distribuidor Commodore y entérese de su precio. Descubrirá qué fácil es poseer el mejor ordenador personal del mundo. El más vendido.





Microelectrónica y Control, s. a. *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid*
Unico representante de Commodore en España.